

UNIVERSIDAD POLITÉCNICA SALESIANA

FACULTAD DE INGENIERÍAS

SEDE QUITO-CAMPUS SUR

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN TELEMÁTICA

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA
EL ACCESO A LA INFORMACIÓN DE LOS AFILIADOS Y
EJECUTIVOS DE LA EMPRESA 2GETHERWEB MEDIANTE
UN PORTAL WAP.

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SISTEMAS

RIVAS RECALDE CARLOS HUGO
SOTO ACOSTA VERÓNICA ELIZABETH

DIRECTOR. ING. FRANKLIN HURTADO LARREA

Quito, ENERO 2011

DECLARACIÓN

Nosotros, Rivas Recalde Carlos Hugo y Soto Acosta Verónica Elizabeth, declaramos bajo juramento que el trabajo aquí es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Rivas Recalde Carlos Hugo

Soto Acosta Verónica Elizabeth

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Rivas Recalde Carlos Hugo y Soto Acosta Verónica Elizabeth bajo mi dirección.

Ing. Franklin Hurtado
Director de Tesis

AGRADECIMIENTO

Estoy consciente de que el gran amor y los grandes logros requieren grandes riesgos, este proyecto me ha servido para darme cuenta de lo inmensamente agradecido que estoy hacia aquellas personas que con su amor y paciencia, directa o indirectamente apoyaron con un granito de arena, me faltarían hojas para expresar mi cariño y gratitud.

Gracias a todos mis amigos, que día a día presionaron para que cumpliera mi meta (nany, mauge, vane, vero, chinita, lucho, perrin y mas); otra persona muy importante que llego casi al final de este camino, que no por estar en el último tramo deja de perder importancia, ya que en poco tiempo me ha demostrado paciencia y he sentido su infinito apoyo, de corazón muchas gracias Dennis.

Gracias a mi compañerita de tesis Verillo... una excelente persona de un inmenso corazón y con un sentido de responsabilidad increíble, sin ti Verillo esta tesis no sería nada. La colega perfecta con la cual este largo camino fue bastante llevadero.

Gracias a mi director de tesis Ing. Franklin Hurtado que nos dio las pautas indispensables para estructurar y comprender de manera clara todo el sistema, que además de ser nuestro tutor llego a ser un gran amigo.

Mi tía Berthy, Dan y Emily desde la distancia siempre estuvieron atentos y preocupados por mi tesis, les quiero mucho.

Ivita mi segunda mama, como olvidar sus abrazos al día siguiente de trasnocharme y siempre mencionando palabras que me daban fuerza para seguir.

Mi pequeña genio (Dianita) estoy tan orgulloso de ti y mucho de todo esto te lo dedico y para mi ñaño Edu mi compañero de siempre gracias por tu eterno apoyo, Mis papis que son las personas por las cuales estoy donde estoy y que sin su incondicional soporte me hubiese faltado energías para concluir con esta etapa de mi vida. LES AMO. **¡Cuán grande riqueza es, aun entre los pobres, el ser hijo de unos buenos padres!.**

Carlos H. Rivas Recalde

AGRADECIMIENTO

Un gran sueño concluyó, un gran camino que tuvo mucho sacrificio y muchas alegrías.. un gran sueño que no hubiera sido posible sin todas esas personas que estuvieron ahí, apoyándome, dándome ánimos y energías para seguir adelante para cumplir mi gran meta.

Primeramente quiero agradecer a mi Dios, gracias por todos los días vividos en mi querida Universidad, por todo lo recibido en esta linda etapa, que me ayudó en todos los aspectos de mi vida a crecer.

Mis amados papitos.. Luis y Jenny que siempre me ayudaron a salir adelante con todo su apoyo, con su gran amor y paciencia.. Mis hermanos Luis y Marcelo, que siempre han estado acompañándome en todo momento.. Mis lindas abuelitas, mis mamitas que siempre fueron la guía y el impulso para llegar a donde estoy.. Gracias gracias por todo.. Los amo infinitamente!!!

Mi grandiosa familia que siempre estuvo pendiente en todo momento, desde el inicio hasta el final de mi vida universitaria, a mis tías Fanny, Chio, Maggy, mis tios Carlos, mi ñaño Rober y mi ñaño Pancho... mis tios Gladys y Rober.. que fueron un gran apoyo cuando más lo necesite.. Gracias

Mis amigos.. gracias chicos porque apoyarme y presionarme para seguir adelante.. gracias por estar ahí Gabyta, Alex, Mili, gracias por todos los hermosos momentos que compartimos en la universidad.. y que hasta ahora los seguimos compartiendo.

Jhon, gracias por darme el impulso para seguir y no decaer... para seguir creciendo en todo aspecto, gracias por el apoyo en todo este tiempo y por el ánimo que me das para seguir adelante en otros caminos.

Mi querido primo adorado Mario, sin ti, jamás hubiese sido salesiana, gracias por estar ahí.

Mi querido tutor Franklin Hurtado, que llego a convertirse en un gran amigo, gracias por todas las pautas dadas para llegar a terminar este gran proyecto, por todas las enseñanzas que nos sirvieron para ser un grandes profesionales.

No puedo dejar de agradecer a mi compañerillo de luchas, mi compañerillo de tesis, definitivamente un gran ser humano y amigo, gracias mi querido Rivas, llavecita parce, Amigo, sin ti no hubiese sido posible terminar esta tesis, gracias por los buenos momentos que hicieron de este sueño, una aventura divertida muy llevadera.. gracias amigui!!! Te quiero.

Gracias totales a todos por ser parte de mi vida y ser la energía que me ha permitido llegar a donde estoy hoy.

Verónica E. Soto Acosta

DEDICATORIA

Este proyecto se lo dedico a mi mama, mi papa, mi ñaña Dianita, mi mami Ivita, mi ñaño Jose Eduardo, mi Tía Berthy, mi primita Emily y a mi tio Dan.

Carlos H. Rivas Recalde

DEDICATORIA

Este gran proyecto quiero dedicárselo a todas esas personas que siempre siempre estuvieron ahí, dándome ánimos y fuerza para salir adelante. Mi gran motor que me mueve, mis hermosa familia.. sin ellos jamás hubiese sido posible terminar este gran sueño que empezó hace ya 6 años. A mis lindos padres Luis y Jenny que siempre estuvieron ahí con su ejemplo y apoyo.. a mis adorados hermanos Luis y Marcelo, a mis abues.. a mi Mami Tere y mi Mami Lola... a todos ustedes va dedicado este gran sueño que gracias a mi Dios se llegó a cumplir... simplemente GRACIAS!!!!!!

Verónica E. Soto Acosta

CONTENIDO

DECLARACIÓN

CERTIFICACIÓN

AGRADECIMIENTOS

DEDICATORIA

CAPÍTULO I: DATOS INFORMATIVOS DEL PROYECTO DE TESIS

1.1. PLANTEAMIENTO DEL PROBLEMA	1
1.2. OBJETIVOS DEL PROYECTO	2
1.2.1. OBJETIVO GENERAL	2
1.2.2. OBJETIVOS ESPECÍFICOS	2
1.3. JUSTIFICACIÓN DEL PROYECTO	2
1.4. ALCANCE	3
1.5. METODOLOGIA	6
1.5.1. GESTIÓN DE PROYECTO	6
1.5.2. GESTIÓN DE DISEÑO Y CONSTRUCCIÓN	7
1.5.3. GESTIÓN DE DISEÑO Y CONSTRUCCIÓN	8

CAPITULO II: MARCO TEÓRICO

2.1. INTRODUCCION 2GETHERWEB	9
2.1.1. MISIÓN	9
2.1.2. VISIÓN	9
2.1.3. OBJETIVO DE TECNOLOGÍA	9
2.2. REDES SOCIALES DE INTERNET	10
2.2.1. REDES SOCIALES EN LOS NEGOCIOS	11
2.3. PROYECTOS Y SUS AMBITOS	12
2.3.1. INTRODUCCIÓN	12
2.4. REDES NEURONALES	13
2.4.1. FUNCIONAMIENTO	13
2.4.2. VENTAJAS QUE OFRECEN LAS REDES NEURONALES	14
2.5. PMP	15
2.5.1. ¿QUES PMP Y PMBOK(Project management book of knowledge)?	15
2.5.2. IMPORTANCIA	15
2.5.3. PROCESOS	15
2.5.4. FASES	16
2.5.5. ENTREGABLES	17
2.5.6. ESTRUCTURA DE TRABAJO	17
2.6. RUP(Rational Unified Process)	18

2.6.1. ¿QUE ES RUP?	18
2.6.2. FASES DE RUP	19
2.6.2.1. Inicio	19
2.6.2.1.1. Entregables de esta fase	19
2.6.2.2. Elaboración	20
2.6.2.2.1. Entregables de esta fase	20
2.6.2.3. Construcción	20
2.6.2.3.1. Entregables de esta fase	20
2.6.2.4. Transición	21
2.6.2.4.1. Entregables de esta fase	21
2.6.3. ESTRUCTURA	21
2.7. ERS (Especificación de Requerimientos del Software)	22
2.7.1. ¿PARA QUÉ SIRVE UN ERS?	22
2.7.2. CLASIFICACION	23
2.7.2.1. Requerimientos funcionales	23
2.7.2.2. Requerimientos no funcionales	24
2.8. MODELO DE NEGOCIO	25
2.9. DIAGRAMA DE PROCESOS	25
2.9.1. DIAGRAMA DE PROCESOS TECNICOS	26
2.9.2. DIAGRAMA DE PROCESOS ORGANIZACIONAL	26
2.10. BPMN (Business Process Modeling Notation)	27
2.10.1. ¿QUE ES BPMN?	27
2.10.2. ¿POR QUÉ ES IMPORTANTE BPMN?	27
2.10.3. FUNDAMENTOS DE BPMN	28
2.10.3.1. Objetos de flujo	28
2.10.3.2. Objetos conectores	30
2.10.3.3. Swimlanes (canales)	31
2.10.3.4. Artefactos	33
2.11. UML (Unified Modeling Language)	34
2.11.1. INTRODUCCION	34
2.11.2. DIAGRAMAS	35
2.11.2.1. Diagramas de Estructura	35
2.11.2.1.1. Diagrama de clases	35
2.11.2.1.2. Diagrama de componentes	37
2.11.2.2. Diagramas de Comportamiento	37
2.11.2.2.1. Diagrama de actividades	38
2.11.2.2.2. Diagramas de casos de uso	38
2.11.2.3. Diagramas de Interacción	38
2.11.2.3.1. Diagrama de secuencia	38
2.11.3. CASO DE USO	40

2.11.3.1.	Definiciones básicas	41
2.11.3.1.1.	Actores	41
2.11.3.2.	Tipos de relaciones	41
2.11.3.2.1.	Inclusión (include o use)	41
2.11.3.2.2.	Extensión (Extend)	42
2.11.3.3.	Normas de aplicación	43
2.11.3.4.	Ventajas	44
2.11.3.5.	Limitaciones	45
2.12.	SERVIDOR WEB	45
2.12.1.	¿QUE ES UN SERVIDOR WEB?	45
2.12.2.	FUNCIONAMIENTO	46
2.13.	MySQL	48
2.13.1.	ARQUITECTURA DE MySQL	48
2.14.	ASP	49
2.14.1.	¿QUE ES ASP?	49
2.14.2.	DESCRIPCIÓN	50
2.14.3.	VERSIONES	50
2.15.	JAVASCRIPT	51
2.15.1.	¿QUÉ ES JAVASCRIPT?	51
2.16.	ARQUITECTURA DE SOFTWARE	52
2.16.1.	INTRODUCCION	52
2.16.2.	RESEÑA HISTORICA	53
2.16.3.	ARQUITECTURA	53
2.16.4.	ARQUITECTURAS MÁS COMUNES	54
2.17.	WAP (Wireless Application Protocol)	55
2.17.1.	¿QUÉ ES LA TECNOLOGÍA WAP?	55
2.17.2.	ARQUITECTURA WAP	56
2.17.2.1.	Capa De Aplicación (Wae)	57
2.17.2.2.	Capa De Sesión (Wsp)	57
2.17.2.3.	Capa De Transacción (Wtp)	58
2.17.2.4.	Capa de seguridad (wtls) y protocolo ssl y set	58
2.17.2.5.	Capa De Transporte (Wdp)	59
2.17.3.	FUNCIONAMIENTO	59
2.18.	GENERACION MOVIL	60
2.18.1.	HISTORIA DE LA TELEFONIA CELULAR	60
2.18.2.	PRIMERA GENERACION 1G	61
2.18.3.	SEGUNDA GENERACION 2G	61
2.18.4.	TERCERA GENERACION 3G	62
2.18.5.	CUARTA GENERACION 4-G	63

2.19. TECNOLOGIAS DE COMUNICACIÓN INALAMBRICA	64
2.19.1. RED INALAMBRICA	64
2.19.1.1. Categorías	64
2.19.1.2. Tipos	64
2.19.1.2.1. Wireless Personal Area Network	65
2.19.1.2.2. Wireless Local Area Network	65
2.19.1.2.3. Wireless Metropolitan Area Network	65
2.19.1.2.4. Wireless Wide Area Network	66
2.20. BIZAGI	66
2.20.1. ¿QUE ES BIZAGI?	66
2.21. TIPOS DE PRUEBA	68
2.21.1. AGUANTE (STRESS TESTING)	68
2.21.2. PRUEBAS DE FUNCIONALIDAD	68
2.21.3. PRUEBAS DE CARGA	68
2.22. TIPOS DE MANUALES	69
2.22.1. ¿QUE ES UN MANUAL?	69
2.22.2. MANUAL DE USUARIO	69

CAPITULO III: ANALISIS Y DISEÑO DEL PROYECTO

3.1. REQUISITOS FUNCIONALES	71
3.1.1. REQUISITOS FUNCIONALES ADMINISTRADOR	71
3.1.2. REQUISITOS FUNCIONALES USUARIO	72
3.2. MODELO DE NEGOCIO	73
3.2.1. MODELO DE NEGOCIO 2GETHERWEB COMPLETO	74
3.2.2. CADENA DE VALOR DETALLADA	74
3.2.2.1. COMERCIALIZACIÓN	74
3.2.2.2. EMPAREJAMIENTO Y ENLACE	75
3.2.2.3. COMUNICACIÓN DE RESULTADOS	75
3.2.2.4. RESULTADOS DE NEURONEER	76
3.2.2.5. PRESTACIÓN DE SERVICIOS DE 2GETHERWEB	76
3.2.2.6. INICIO DE SESION Y VALIDACIÓN DE USUARIOS	77
3.2.2.7. INFORMACIÓN Y COMUNICACIÓN	77
3.2.2.8. VISUALIZACIÓN DE NOTICIAS	78
3.2.2.9. VISUALIZACIÓN DE ESTADO DE CUENTA	78
3.2.2.10. ASESORÍA Y SOPORTE AL AFILIADO	79
3.2.2.11. MONITOREO Y ADMINISTRACIÓN DEL PORTAL Y LOS AFILIADOS	81
3.2.2.12. MONITOREO DE INDICADORES	80
3.2.2.13. ESTADÍSTICAS	81

3.2.2.14.	GESTIÓN DEL VECTOR	81
3.2.2.15.	INTERACCIÓN CON AFILIADOS	82
3.2.2.16.	RESPUESTA DE TICKETS	82
3.2.2.17.	ORQUESTACIÓN DE PROCESOS	82
3.3.	DIAGRAMACIÓN	83
3.3.1.	DIAGRAMA DE CASOS DE USO	84
3.3.1.1.	INICIO DE SESION USUARIO	84
3.3.1.2.	MENÚ AFILIADO	85
3.3.1.3.	PERFIL	86
3.3.1.4.	NOTICIAS RECIBIDAS	87
3.3.1.5.	TICKETS DE SERVICIO	88
3.3.1.6.	RESULTADOS NEURONEER	90
3.3.1.7.	VITRINA DE LÍDERES	91
3.3.1.8.	CAMBIO DE CONTRASEÑA	92
3.3.1.9.	ESTADÍSTICAS	92
3.3.1.10.	RESPUESTA DE TICKETS	93
3.3.1.11.	VECTOR	95
3.3.2.	DIAGRAMA DE ACTIVIDADES	96
3.3.2.1.	INICIAR SESION USUARIO	96
3.3.2.2.	VISUALIZAR PERFIL	96
3.3.2.3.	VISUALIZAR NOTICIAS	97
3.3.2.4.	CREAR TICKETS	97
3.3.2.5.	VISUALIZAR NEURONEER	98
3.3.2.6.	VISUALIZAR VITRINA DE LÍDERES	98
3.3.2.7.	CAMBIAR DE CONTRASEÑA	99
3.3.2.8.	VISUALIZAR ESTADÍSTICAS	99
3.3.2.9.	RESPONDER TICKET	99
3.3.2.10.	GESTIONAR VECTOR	100
3.3.3.	DIAGRAMA NAVEGACIONAL	100
3.3.3.1.	DIAGRAMA NAVEGACIONAL USUARIO AFILIADO	100
3.3.3.2.	DIAGRAMA NAVEGACIONAL ADMINISTRADOR	101
3.3.4.	DIAGRAMA DE INTERFAZ	101
3.3.4.1.	INICIO SESION	101
3.3.4.2.	MENÚ USUARIO	102
3.3.4.3.	NEURONEER	102
3.3.4.4.	VITRINA DE LÍDERES	103
3.3.4.5.	PERFIL	103
3.3.4.6.	RESUMEN NOTICIAS	104
3.3.4.7.	NOTICIAS	104
3.3.4.8.	CREAR TICKETS	105

3.3.4.9. RESPUESTA DE TICKETS	105
3.3.4.10. NUEVO TICKET	106
3.3.4.11. CAMBIO DE CONTRASEÑA	106
3.3.4.12. ESTADÍSTICAS	107
3.3.4.13. RESPUESTA DE TICKETS	107
3.3.4.14. GESTIÓN VECTOR	108
3.3.5.DIAGRAMA DE CLASES	109
3.3.6.DIAGRAMA DE SECUENCIA	109
3.3.7.DIAGRAMA DE DESPLIEGUE	110

CAPÍTULO IV: IMPLEMENTACIÓN E IMPLANTACIÓN

4.1. ESTÁNDARES DE PROGRAMACIÓN	111
4.1.1.COMENTARIOS	111
4.1.2.VARIABLES	111
4.2. EXPLICACIÓN DE CÓDIGO	112
4.2.1.CLASES	112
4.2.2.MÉTODOS	113
4.3. DICCIONARIO DE DATOS	115
4.4. REGISTRO DE CONSULTAS	136
4.5. PRUEBAS	142
4.5.1.PRUEBA DE FUNCIONALIDAD	142
4.5.1.1. READY.MOBI	142
4.5.1.1.1. Login de acceso	142
4.5.1.1.2. Página Principal	143
4.5.1.1.3. Page Test	144
4.5.1.1.4. Site Test	145
4.5.1.1.5. Resultados De Los Test Realizados	146
4.5.1.1.6. Resultados De Los Sitios	146
4.5.1.1.7. Resultado Individual: Mobileadmin	147
4.5.1.1.8. Resultado Individual: Mobile	147
4.5.2.PRUEBAS DE CARGA	148
4.5.2.1. Transferred Data & System & CPU Load	148
4.5.2.2. Reporte RAMP – Open Requests and Transferred Data	149
4.5.2.3. Reporte TIME – Click Time, Hits/s, Users/s (all URLs)	149
4.5.2.4. Server and User Bandwidth	150

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES	151
-------------------	-----

ANEXO A

ANEXO B

ANEXO C

ANEXO D

ÍNDICE DE FIGURAS

CAPITULO II: MARCO TEÓRICO

Fig. 2.1 Redes Sociales	10
Fig. 2.2 . Ejemplo de proyecto de una sola fase.	16
Fig. 2.3 Ejemplo simple de un EDT	18
Fig. 2.4 Fases de RUP	22
Fig. 2.5 Fases de RUP	26
Fig. 2.6 Objetos de flujo – Evento	29
Fig. 2.7 Objetos de flujo – Actividad	29
Fig. 2.8 Objetos de flujo – Gateway	30
Fig. 2.9 Objetos de flujo – Conectores	30
Fig. 2.10 Objetos de flujo – Canales	31
Fig. 2.11 Ejemplo de Canales	32
Fig. 2.12 Ejemplo de Canales	32
Fig. 2.13 Artefactos	33
Fig. 2.14 Ejemplo de modelado de procesos	34
Fig. 2.15 Jerarquía de los diagramas UML 2.0, mostrados como un diagrama de clases	35
Fig. 2.16 Ejemplo de Diagrama de Clases	35
Fig. 2.17 Ejemplo de Diagrama de Secuencia	39
Fig. 2.18 Casos de Uso	40
Fig. 2.19 Ejemplo Include	41
Fig. 2.20 Ejemplo Extend	42
Fig. 2.21 Arquitectura Lógica	48
Fig. 2.22 Arquitectura Monolítica: aplicación y datos en un sólo cliente aislado.	54
Fig. 2.23 Arquitectura Cliente/Servidor	54
Fig. 2.24 Arquitectura Tres Capas	55
Fig. 2.25 Arquitectura WAP	57
Fig. 2.26 Funcionamiento WAP	60
Fig. 2.27 Primera Generación	61
Fig. 2.28 Segunda Generación	62
Fig. 2.29 Tercera Generación	62
Fig. 2.30 Cuarta Generación	63
Fig. 2.31 Tipos de Cobertura	64
Fig. 2.32 Ejemplo Uso Bizagi	67

CAPITULO III: ANALISIS Y DISEÑO DEL PROYECTO

Fig.3.1 Modelo de Negocios 2GETHERWEB	74
Fig.3.2 Cadena de Valor – Comercialización.	75
Fig.3.3 Cadena de Valor – Emparejamiento y enlace.	75
Fig.3.4 Cadena de Valor – Comercialización de resultados.	76
Fig.3.4 Cadena de Valor – Resultados de Neuroneer.	76
Fig.3.5 Cadena de Valor – Prestación de servicios de 2GETHERWEB.	77
Fig.3.6 Cadena de Valor – Login y validación de usuarios	77
Fig.3.7 Cadena de Valor – Información y comunicación.	78
Fig.3.8 Cadena de Valor – Visualización de noticias.	78
Fig.3.9 Cadena de Valor – Visualización de estado de cuenta.	79
Fig.3.10 Cadena de Valor – Asesorías y soporte al afiliado	80
Fig.3.11 Cadena de Valor – Monitoreo y administración del portal y los afiliados.	80
Fig.3.12 Cadena de Valor – Monitoreo de indicadores.	81
Fig.3.13 Cadena de Valor – Estadísticas	81
Fig.3.14 Cadena de Valor – Gestión del vector.	82
Fig.3.15 Cadena de Valor – Interacción con afiliados.	82
Fig.3.16 Cadena de Valor – Respuesta de tickets.	82
Fig.3.17 Cadena de Valor – Orquestación de procesos.	83
Fig.3.18 Caso de Uso –Logueo usuario	84
Fig.3.19 Caso de Uso – Menú Afiliado	85
Fig.3.20 Caso de Uso –Perfil	86
Fig.3.21 Caso de Uso – Noticias Recibidas	87
Fig.3.22 Caso de Uso – Tickets de servicio	88
Fig.3.23 Caso de Uso – Neuroneer	90
Fig.3.24 Caso de Uso – Vitrina de Líderes	91
Fig.3.25 Caso de Uso – Cambio de contraseña	92
Fig.3.26 Caso de Uso – Estadísticas	92
Fig.3.27 Caso de Uso – Respuesta de Tickets	93
Fig.3.28 Caso de Uso – Vector	95
Fig.3.29 Actividades – Loguear usuario.	96
Fig.3.30 Actividades – Visualizar perfil.	96
Fig.3.31 Actividades – Visualizar noticias	97
Fig.3.32 Actividades – Crear tickets	97
Fig.3.33 Actividades – Visualizar Neuroneer	98
Fig.3.34 Actividades – Vitrina de Líderes	98
Fig.3.35 Actividades – Cambio de contraseña	
Fig.3.36 Actividades – Visualizar estadísticas	99

Fig.3.37 Actividades – Responder ticket	99
Fig.3.38 Actividades – Gestionar Vector	99
Fig.3.39 Navegacional – Usuario afiliado.	100
Fig.3.40 Navegacional – Usuario administrador	100
Fig.3.41 Interfaz Abstracta – Logueo.	101
Fig.3.43 Interfaz Abstracta – Menú	101
Fig.3.44 Interfaz Abstracta – Neuroneer	102
Fig.3.45 Interfaz Abstracta – Vitrina de Líderes	102
Fig.3.46 Interfaz Abstracta – Perfil	103
Fig.3.47 Interfaz Abstracta – Resumen noticias	103
Fig.3.48 Interfaz Abstracta – Noticias.	104
Fig.3.49 Interfaz Abstracta – Crear tickets.	105
Fig.3.50 Interfaz Abstracta – Revisión de tickets	105
Fig.3.51 Interfaz Abstracta – Nuevo ticket.	106
Fig.3.52 Interfaz Abstracta – Cambio contraseña	106
Fig.3.53 Interfaz Abstracta – Estadísticas	107
Fig.3.54 Interfaz Abstracta – Respuesta de tickets	107
Fig.3.55 Interfaz Abstracta – Respuesta de tickets Detalle	108
Fig.3.56 Interfaz Abstracta – Gestión Vector.	108
Fig.3.57 Diagrama de Clases	109
Fig.3.58 Diagrama de Secuencia	109
Fig.3.59 Diagrama de Despliegue	110

CAPÍTULO IV: IMPLEMENTACIÓN E IMPLANTACIÓN

Fig. 4.1 Login de acceso a mobiReady	142
Fig. 4.2 Pagina Principal	143
Fig. 4.3 Visualización del Page Test	144
Fig. 4.4 Visualización del Site Test	145
Fig. 4.5 Visualización de página Site	146
Fig. 4.6 Visualización de Resultados	146
Fig. 4.7 Detalle Resultado Individual Mobileadmin	147
Fig. 4.8 Detalle Resultado Individual mobile	147
Fig. 4.9 Reporte de carga 1	148
Fig. 4.10 Reporte de carga 2	149
Fig. 4.11 Reporte de carga 3	149
Fig. 4.12 Reporte de carga 4	150

ABSTRACT

Hoy en día, las redes sociales de Internet han tomado mucha fuerza, por lo que muchas empresas han decidido implementarlas para generar mayor productividad y ganancia, y de esta manera captar más clientes. Por esta razón 2GETHERWEB decide formar una red social mediante la cual, muchas personas puedan proponer sus ideas de negocios, fomentando el crecimiento de los mismos mediante el enlace que Neuroneer, un software de marca registrada, realiza entre los contactos que tiene cada afiliado. Debido al gran impacto que tuvo 2GETHERWEB, se decide implementar una aplicación móvil, que contenga las funcionalidades más importantes de su página WEB y de esta manera brindar a sus usuarios un acceso rápido que genere la información que ellos necesitan en ese instante. Además se decide crear el portal para que también el Administrador pueda responder a los requerimientos que tienen los clientes en tiempo real. El proyecto fue realizado basándose en la metodología RUP, utilizando diagramas UML, los mismos que fueron muy útiles para el análisis, diseño e implementación del portal WAP. El resultado obtenido cumplió con las metas establecidas inicialmente, satisfaciendo así las expectativas tenidas por la empresa.

PRESENTACIÓN

2GETHERWEB pensando en ofrecer un mejor servicio, decidió crear una aplicación móvil para el acceso de sus afiliados y así proveer la información necesaria de forma rápida, desde cualquier lugar y en cualquier momento, el mismo que muestra contenido primordial necesario para el manejo de cada afiliado, además permite que los usuarios administradores puedan responder a las inquietudes que tengan sus afiliados mediante el manejo de tickets.

El proyecto fue analizado utilizando metodología RUP, el mismo que nos permitió generar todos los diagramas necesarios para desarrollar la aplicación móvil requerida, implementado así un producto que cumple con todos los requerimientos propuestos por 2GETHERWEB.

CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA.

En el área de los negocios es cada vez más notoria la necesidad de prestar servicios directos y brindar información a los usuarios de manera rápida, eficaz y concisa.

2GETHERWEB es una Red Profesional de Soluciones Interdisciplinarias compuesta por un gigante almacén virtual que acoge Necesidades, Intereses, Ofertas, Demandas e Informaciones de Individuos, Empresas e Instituciones a nivel mundial para lo cual cuenta con un portal WEB, el mismo que brinda muchas facilidades a sus usuarios.

No obstante, el principal problema o inconveniente que tiene la empresa es la imposibilidad de acceso constante a la información de manera móvil, ya que, por lo general se dispone de medios remotos estáticos como laptops e inclusive computadoras fuera del entorno laboral lo cual es un serio limitante para consultar de forma inmediata los datos que sean requeridos tanto por usuarios o ejecutivos.

Esta empresa crece diariamente debido a que la demanda de acceso a la información es constante, por lo cual, se necesita un monitoreo permanente, gestión inmediata de los estados de cuenta, además de que, los datos que se solicitan sean presentados con un tiempo de respuesta acorde a los usuarios que se encuentran dispersos alrededor del mundo y que por cuestiones de diferencias de horarios no tienen la pronta atención a sus requerimientos; este ambiente de indisponibilidad también es aplicable para los ejecutivos o administradores que por motivos de viajes de negocios o emergencias tienen la dificultad de llevar un seguimiento inmediato de las actividades que se encuentran en demanda de finalización.

1.2. OBJETIVOS DEL PROYECTO.

1.2.1. OBJETIVO GENERAL.

Implementar un portal WAP para facilitar el acceso constante a la información requerida por los usuarios y ejecutivos de la empresa 2GETHERWEB ampliando significativamente la capacidad de difundir su información y así propiciar un crecimiento rápido de la organización.

1.2.2. OBJETIVOS ESPECÍFICOS.

- Desarrollar un portal WAP para dispositivos móviles utilizando las herramientas y metodologías especificadas.
- Gestionar el proyecto de este sistema en base a PMP (Project Management Professional) una metodología dada por el PMI (Project Management Institute).
- Utilizar lo menos posible los activos de la organización, para hacer de este sistema algo conveniente no solo por su eficacia, sino, por su rentabilidad para la misma.
- Desarrollar un manual de usuario y de administración del sistema.

1.3. JUSTIFICACIÓN DEL PROYECTO.

Con la finalidad de abarcar mayor cantidad de clientes y de estar en carrera constante con la tecnología; 2GETHERWEB decidió implementar por medio de un portal WAP las funcionalidades más importantes de su página WEB.

Con la búsqueda de la satisfacción de sus afiliados en tres ámbitos principales; Negocios; Salud y Conocimiento. Para lograr este objetivo el uso de un portal WAP es una decisión válida y acertada con mayor probabilidad de compatibilidad y éxito, ya que, es uno de los medios más ágiles de acceso a la información que existe en el momento.

Mediante un dispositivo móvil se cumple con la finalidad de que el cliente tenga la información en la palma de su mano sin la necesidad de estar en su escritorio frente a una computadora.

La empresa 2GETHERWEB quiere que sus clientes estén enterados y tengan acceso rápido a sus cuentas en la página, las últimas noticias concernientes a los negocios, además de mostrar los contactos para posibilidades de nexos y nuevos emprendimientos.

Al igual que los clientes; los ejecutivos de 2GETHERWEB se verán beneficiados mediante la consola administrativa; con esta se podrán manejar procesos primordiales y finalización de actividades.

Es por lo mencionado acerca de este portal WAP que se hace factible su realización, debido que; tanto clientes como ejecutivos de la empresa tendrán un beneficio tecnológico y una inmejorable manera de expandirse en el mercado.

1.4. ALCANCE.

Para desarrollar los módulos que requerimos implementar, se usarán las herramientas necesarias para poder ejecutarlo con la mayor prontitud, estas son:

- Para el desarrollo de base de datos se utilizará SQL Server, debido a que se va a gestionar un estimado de 1'000.000 de registros solamente para la información de usuarios o afiliados, igualmente se considera una cantidad no menor a 500.000 registros entre actividades y subactividades que se realizaran de manera simultánea.
- Para desarrollar los módulos se utilizara ASP, debido a que será aplicado para una plataforma móvil, para así obtener los resultados que se desean de manera oportuna y rápida, ya que con esto se optimizarán recursos para que los usuarios del Portal WAP tengan un mejor acceso.

El proyecto será implementado desarrollando un Portal WAP, debido a que nos permite utilizar aplicaciones de manera móvil, como se trata de la especificación de un entorno de aplicación y de un conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos se pueden utilizar para acceder a: correo electrónico, grupo de noticias y otros. De esta manera, todos los usuarios de 2GETHERWEB pueden acceder de manera inalámbrica e inmediata a los servicios que se ofrecen.

Portal WAP “2GETHERMOVIL”

Este portal constaría de los siguientes módulos:

- **Módulo del afiliado (suscriptor)**

Líneas Funcionales (Submódulos).

1) Ingreso con usuario y contraseña.

- Ingreso en 2 idiomas: Inglés y Español.

2) Información de Noticias.

- Visualización de las noticias.
- Búsqueda mediante un índice de título.

3) Información de la cuenta del Afiliado (Estado de Cuenta).

- Características de la cuenta del Usuario, donde se mostrará un pequeño estado de cuenta detallado y preciso sobre los movimientos que el afiliado ha realizado, este estado será actual y mostrará el tipo de rendimiento que tiene, además del estado de amigos con el que cuenta.

4) Resultados de Neuroneer.

- Visualización y seguimiento de resultados obtenidos, donde podrá hacer el seguimiento de todos los eventos e intereses generados.

5) Mensajería Rápida

- Los afiliados podrán enviar invitaciones a sus amigos utilizando mensajes de envío rápido donde se informara los beneficios de utilizar y pertenecer a 2GETHERWEB.

6) Respuestas de Tickets

- Tickets de Atención

- **Módulo administrativo**

Líneas Funcionales (Submódulos).

1) Ingreso con usuario y contraseña del administrador.

2) Información de Estadísticas.

Se presentarán estadísticas claras y concisas sobre el desarrollo del portal y de los afiliados, para ello se utilizará:

- Estadísticas con gráficos
- Estadísticas mediante tablas

3) Gestión de E-mails.

Se realizará el envío de mails para todos los ejecutivos de la empresa.

4) Mensajería Interna.

- Se manejará el envío de mensajes de texto a todos los ejecutivos donde se informará sobre mails recibidos y además el mensaje contendrá un resumen del mail, lo que facilitará una respuesta más rápida a los requerimientos que se tengan en ese instante.

5) Gestión de Vector (Panel Personal)

- Con el uso del vector, se mantendrá un control sobre los procesos principales que se realizan internamente en la empresa, donde se

visualizará los estados por los que se encuentran dichos procesos y quien los ejecuta.

6) Gestión de Alarmas.

- Con este submódulo los ejecutivos tienen una forma de aviso de las actividades que se están realizando con el vector, su secuencia de acción, notificaciones de terminación de procesos, etc.; esto se realizara mediante correo electrónico,

Los módulos de software serán diseñados para que funcionen en portales móviles, específicamente para Blackberry y iPhone, los mismos que dependerán de las necesidades de cada afiliado.

El programa no manejara los siguientes puntos:

- Creación de nuevos usuarios
- Modificación del perfil del afiliado
- Registro de los servicios de 2GETHERWEB

1.5. METODOLOGÍA.

1.5.1. GESTIÓN DE PROYECTO.

Es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos.

Para el proyecto de tesis a desarrollar utilizaremos PMI (Project Management Institute). El PMI es una asociación profesional sin ánimo de lucro fundada 1969 cuya misión es mejorar la práctica, la ciencia y la profesión de la gestión de proyectos a través del mundo con una actitud consciente y proactiva con el objetivo de que las organizaciones adopten, valoren y utilicen la gestión de proyectos y luego le atribuyan a éste, su éxito.

1.5.2. GESTIÓN DE DISEÑO Y CONSTRUCCIÓN.

Para esta parte del desarrollo del proyecto de tesis utilizaremos diferentes opciones para que la gestión sea llevada de la mejor manera:

- **RUP(Proceso Unificado de Rational)**

Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP tiene un conjunto de metodologías adaptables al contexto y necesidades del desarrollo de este proyecto.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

Se establecerá un modelo de fases con el cual se va a llevar a acabo el proyecto, esto es de suma importancia para el ciclo de vida del sistema y para eventos de control. A continuación se delimitará las fases a utilizar:

- Iniciación.
- Elaboración.
- Construcción.
- Transición.

- **UML(Lenguaje Unificado de Modelado)**

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Lo utilizaremos para la diagramación ya que se aplica y se acopla fácilmente en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software.

- *BPMN*(Notación para el Modelado de Procesos de Negocio)

Es una notación gráfica estandarizada que permite el modelado de procesos de negocio; el principal objetivo de BPMN es proveer una notación estándar que sea fácilmente leíble y entendible por parte de todos los involucrados e interesados del negocio.

Se tomaran muy en cuenta los estándares propuestos por esta notación en la fase de Análisis del Proyecto.

- *OOHDM*(Modelo de diseño de hipermedia orientado a objetos)

Es una mezcla de estilos de desarrollo basado en prototipos, en desarrollo interactivo y de desarrollo incremental. En cada fase se elabora un modelo orientado a objetos conceptual que recoge las características a resaltar en la misma incrementando los resaltados de la fase o fases anteriores.

1.5.3. GESTIÓN INVESTIGATIVA.

Dividiremos este tipo de metodología en dos partes:

- Documental (Científica).

Nos basaremos en la parte bibliográfica del punto de vista institucional y de la ciencia que está siendo utilizada en 2GETHERWEB.

- De Campo.

Se realizara una investigación de campo, es decir, tratar de tomar todos los recursos posibles dentro de la empresa; esto consiste en realizar consultas, entrevistas, pruebas dentro de la empresa del sistema, etc.

CAPITULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN 2GETHERWEB.

2.1.1 MISIÓN

Formar la más grande empresa bioética del universo on-line con el fin de humanar al Internet, creando una multiplicidad de redes de grandiosas oportunidades para el Hombre y sus sociedades.

2.1.2 VISIÓN

Somos 2getherWeb, un mega sistema on-line, que opera con el único cerebro artificial denominado Neuroneer ; un millón de veces más capaz que el cerebro humano..! En tres años seremos los mejores del mundo en la formación de hombres y comunidades interactuantes y con la mayor penetración en el mercado Web de los 5 continentes; habremos concretado más de 220 mil negocios y levantado el Heaven XXI (por ponerlo un nombre) compuesto por las 10,000 personas y empresas globales que liderarán este siglo en todas las profesiones, experticias, manifestaciones, creaciones y realizaciones , con el fin de que sean el ejemplo presente y en tiempo real para los miembros de nuestro sistema.

2.1.3 OBJETIVO DE TECNOLOGÍA

Su principal producto es Neuroneer, un software de marca registrada que permite crear enlaces entre sus afiliados para así establecer los contactos que sean necesarios para que sus proyectos puedan tener apoyo tecnológico y obtener el financiamiento requerido.

2.2 REDES SOCIALES DE INTERNET

Las redes sociales en Internet han ganado su lugar de una manera vertiginosa convirtiéndose en promisorios negocios para empresas, artistas, marcas y sobretodo en lugares para encuentros humanos de carácter sentimental.

“Las Redes son formas de interacción social, definida como un intercambio dinámico entre personas, grupos e instituciones en contextos de complejidad. Un sistema abierto y en construcción permanente que involucra a conjuntos que se identifican en las mismas necesidades y problemáticas y que se organizan para potenciar sus recursos. Una sociedad fragmentada en minorías aisladas, discriminadas, que ha desvitalizado sus redes vinculares, con ciudadanos carentes de protagonismo en procesos transformadores, se condena a una democracia restringida. La intervención en red es un intento reflexivo y organizador de esas interacciones e intercambios, donde el sujeto se funda a sí mismo diferenciándose de otros.”

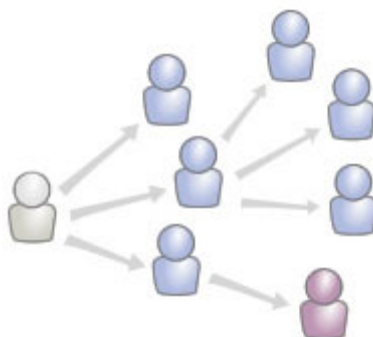


Fig. 2.1 Redes Sociales.

Autor: Marcelo Zamora

Fuente: www.maestrosdelWeb.com, Reseña de Redes Sociales

Por la dinámica actual, las relaciones interpersonales han perdido los espacios y por ende son cada vez menos frecuentes, al menos bajo los medios convencionales; como respuesta a esta coyuntura han ido apareciendo las denominadas redes sociales.

En las redes sociales de Internet tenemos la posibilidad de interactuar con otras personas aunque no las conozcamos, el sistema es abierto y obviamente se va

construyendo con lo que cada suscripto a la red aporta, en este sentido se puede decir que cada nuevo miembro que ingresa, transforma al grupo en otro nuevo. Por ello la red no es la misma si uno de sus miembros deja de ser parte de ella.

Intervenir en una red social empieza cuando un individuo pretende encontrar a otros con quienes compartir sus intereses, preocupaciones o necesidades y aunque no sucediera más que eso, eso mismo ya es mucho porque rompe el aislamiento que suele aquejar a la gran mayoría de las personas, lo cual suele manifestarse en retraimiento y en otras ocasiones en excesiva vida social sin afectos comprometidos.

2.2.1 REDES SOCIALES EN LOS NEGOCIOS.

Vivimos un momento de cambio, observamos cómo se transforma el sistema clásico de producción, es por esto que, las comunidades virtuales en Internet son muy importantes en el ámbito de los negocios.

Desde el punto de vista empresarial las CVI¹ permiten a las organizaciones mejorar su dinámica de trabajo interno, las relaciones con sus clientes o incrementar su eficiencia en procesos.

Durante años se ha puesto el énfasis en que la tecnología, por sí misma, aporta significativamente a que las empresas logren productividad. Esto puede ser más evidente en las compañías cuyos elementos se basan en bienes de equipo, toda vez que está demostrado que cada vez más trabajadores son 'brokers de la información'. Los analistas han dejado de medir los tangibles -la tecnología- para preocuparse por los intangibles -los flujos y gestión de información en las organizaciones-. Y es entonces cuando se percatan de que un empleado no produce más y mejor por tener un magnífico ordenador y el último modelo de Blackberry², sino la capacidad de dicho empleado en usarlos para establecer y

¹ Comunidad Virtual de Internet, ver en Glosario

² Línea de teléfonos inteligentes (smartphones) que integran el servicio de correo electrónico móvil.

explotar al máximo sinergias con sus compañeros de trabajo y, por qué no, colegas de profesión aunque sean de otras empresas o centros de investigación.

"Más y más diversas relaciones sociales y de negocios mejoran las habilidades para la gestión de la información y esto se traduce en incrementos en la productividad".³

2.3 PROYECTOS Y SUS ÁMBITOS

2.3.1 INTRODUCCIÓN

Los proyectos por definición tienen una fecha de inicio y de finalización determinadas, al igual que un alcance, presupuesto, resultados específicos y recursos asignados. Adicionalmente, cada proyecto, por similar que sean las actividades y los alcances, se tornan diferentes porque las circunstancias cambian, y las cosas siempre son distintas cuando se lidia con personas.

Cuando se define el proyecto, se debe asegurar que se tiene un entendimiento claro con el patrocinador del mismo sobre qué se debe alcanzar en el proyecto. En la etapa de planeación se determina cómo se va realizar el trabajo, esto implica entonces, que se debe elaborar un plan para el trabajo. Se utilizan diferentes alcances, acordes al tamaño del proyecto.

Hay dos temas fundamentales en la administración de un contrato: la entrega de los resultados (entregables) y el cumplimiento de las fechas para estas entregas. Para ambos se establecen criterios y requerimientos que permiten controlar el cumplimiento de los dos temas.

El plan de trabajo es solo un entregable. Describe lo que hay que hacer, el orden del trabajo, el esfuerzo requerido y quien está asignado a qué tarea, pero solo

³ Extraído de "Los trabajadores más conectados son los más rentables", por Juan Freire.

representa el mejor estimado de cómo completar el trabajo que queda por hacer en un momento dado de un proyecto.

Los criterios y cumplimientos son definidos por las personas responsables de que el proyecto sea manejado de manera que las partes involucradas tengan una ganancia mutua, es decir, que el beneficio sea imperativo y tenga una alta relevancia en la relación costo/beneficio.

2.4 REDES NEURONALES

2.4.1 FUNCIONAMIENTO

Es un sistema compuesto por un gran número de elementos básicos, agrupados en capas y que se encuentran altamente interconectados; Esta *estructura* posee varias entradas y salidas, las cuales serán entrenadas para reaccionar (*valores O*), de una manera deseada, a los estímulos de entrada (*valores I*). Estos sistemas emulan, de una cierta manera, al *cerebro* humano. Requieren aprender a comportarse y alguien debe encargarse de enseñarles o entrenarles, en base a un *conocimiento* previo del entorno del problema, siempre y cuando el esquema sea el llamado supervisado.

Las redes neuronales son dispositivos inspirados en la funcionalidad de las neuronas biológicas, aplicados al reconocimiento de patrones que las convierten aptas para modelar y efectuar predicciones en sistemas muy complejos.

Es un conjunto de técnicas matemáticas para modelar las conexiones y relaciones entre un conjunto de datos.

Las Redes Neuronales surgieron del movimiento conexionista, que nació junto con la IA ⁴simbólica o tradicional. La IA simbólica se basa en que todo conocimiento se puede representar mediante combinaciones de símbolos, derivadas de otras combinaciones que representan verdades incuestionables o

⁴ Inteligencia Artificial, ver en Glosario.

axiomas. Así pues, la IA tradicional asume que el conocimiento es independiente de la estructura que maneje los símbolos, siempre y cuando la 'máquina' realice algunas operaciones básicas.

2.4.2 VENTAJAS QUE OFRECEN LAS REDES NEURONALES

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Entre las ventajas se incluyen:

- Aprendizaje Adaptativo: capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- Auto-organización: una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- Tolerancia a fallos: la destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
- Operación en tiempo real: los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
- Fácil inserción dentro de la tecnología existente: se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

Otras combinaciones que representan verdades incuestionables o axiomas. Así pues, la IA tradicional asume que el conocimiento es independiente de la estructura que maneje los símbolos, siempre y cuando la 'máquina' realice algunas operaciones básicas entre ellos.

2.5 PMP

2.5.1 ¿QUÉ ES PMP Y PMBOK(Project management book of knowledge)?

La *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK®)* es una norma reconocida en la profesión de la dirección de proyectos. Por norma se hace referencia a un documento formal que describe normas, métodos, procesos y prácticas establecidos. Al igual que en otras profesiones, como la abogacía, la medicina y las ciencias económicas, el conocimiento contenido en esta norma evolucionó a partir de las buenas prácticas reconocidas por profesionales dedicados a la dirección de proyectos, quienes contribuyeron a su desarrollo.

2.5.2 IMPORTANCIA

La creciente aceptación de la dirección de proyectos indica que la aplicación de conocimientos, procesos, habilidades, herramientas y técnicas adecuados puede tener un impacto considerable en el éxito de un proyecto. La *Guía del PMBOK®* identifica ese subconjunto de fundamentos de la dirección de proyectos generalmente reconocido como buenas prácticas. “Generalmente reconocido” significa que los conocimientos y prácticas descritos se aplican a la mayoría de los proyectos, la mayor parte del tiempo, y que existe consenso sobre su valor y utilidad. “Buenas prácticas” significa que se está de acuerdo, en general, en que la aplicación de estas habilidades, herramientas y técnicas puede aumentar las posibilidades de éxito de una amplia variedad de proyectos. Buenas prácticas no significa que el conocimiento descrito deba aplicarse siempre de la misma manera en todos los proyectos; la organización y/o el equipo de dirección del proyecto son responsables de establecer lo que es apropiado para un proyecto determinado.

2.5.3 PROCESOS

Aunque muchos proyectos pueden tener fases con nombres y entregables similares, pocos son idénticos. Algunos proyectos tendrán una sola fase.

Otros, en cambio, pueden constar de muchas. El Gráfico 2.2 muestra un ejemplo de proyecto de tres fases. Normalmente, las diferentes fases tienen una duración o longitud diferente.

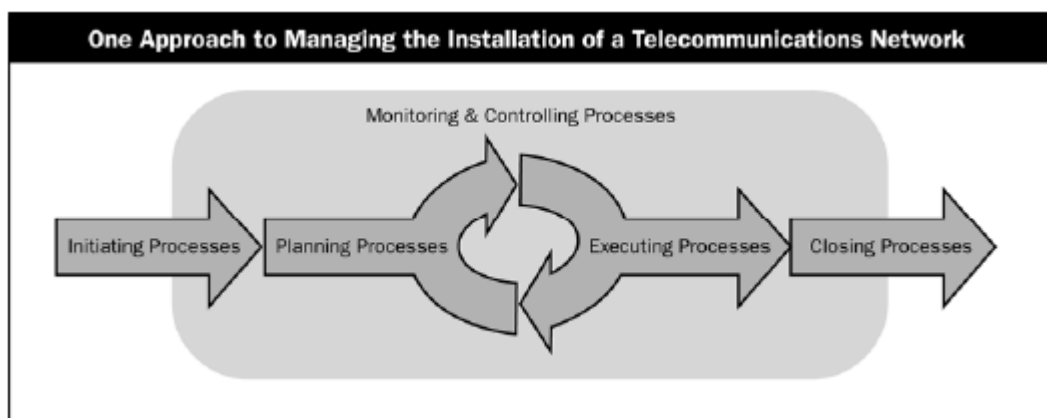


Fig. 2.2 Ejemplo de proyecto de una sola fase

Fuente: PMBOK

No existe una manera única de definir la estructura ideal de un proyecto. Aunque las prácticas comunes de la industria conduzcan con frecuencia a utilizar una estructura preferida, los proyectos en la misma industria, o incluso dentro de la misma organización.

2.5.4 FASES

Las fases del proyecto son divisiones dentro del mismo, donde es necesario ejercer un control adicional para gestionar eficazmente la conclusión de un entregable mayor. Estas suelen completarse de manera secuencial, pero en determinadas situaciones pueden superponerse.

Por su naturaleza de alto nivel, las fases constituyen un elemento del ciclo de vida del proyecto.

Estructurarlas permite dividir al proyecto en subconjuntos lógicos para facilitar su dirección, planificación y control. El número, la necesidad de establecer fases y el

grado de control aplicado dependen del tamaño, la complejidad y el impacto potencial del proyecto. Independientemente de la cantidad de etapas que compongan un proyecto, todas ellas poseen características similares:

- Cuando las fases son secuenciales, el cierre de una de ellas termina con cierta forma de transferencia o entrega del trabajo producido como el entregable. Su terminación representa un punto natural para re-evaluar el esfuerzo en curso y, en caso de ser necesario, para cambiar o terminar el proyecto. Estos puntos se conocen como salidas, hitos, puertas de fase, puntos de decisión, puertas de etapa o puntos de cancelación.
- El trabajo tiene un enfoque único que difiere del de cualquier otra fase. Esto involucra a menudo diferentes organizaciones y conjuntos de habilidades.
- Para alcanzar con éxito el objetivo o entregable principal, se requiere un grado adicional de control. La repetición de procesos a través de los cinco grupos de procesos proporciona ese grado adicional de control y define los límites de la fase.

2.5.5 ENTREGABLES

Es un objeto tangible o intangible que resulta de la ejecución de un proyecto o una parte de él, entonces en este sentido un proyecto se puede componer de varios entregables más pequeños. Como entregable podemos tener: documentos, software, diagramas beneficios, productos varios.

2.5.6 ESTRUCTURA DE TRABAJO

En gestión de proyectos, una **estructura de descomposición del trabajo** o **EDT**, también conocido por su nombre en inglés **Work Breakdown Structure** o **WBS**, es una estructura exhaustiva, jerárquica y descendente formada por los entregables a realizar en un proyecto. La EDT es una herramienta muy común y crítica en la gestión de proyectos.

El propósito de una EDT es documentar el alcance del proyecto. Su forma jerárquica permite una fácil identificación de los elementos finales. Siendo un elemento exhaustivo en cuanto al alcance del proyecto, la EDT sirve como la base para la planificación del proyecto. Todo trabajo a ser hecho en el proyecto debe poder rastrear su origen en una o más entradas de la EDT.

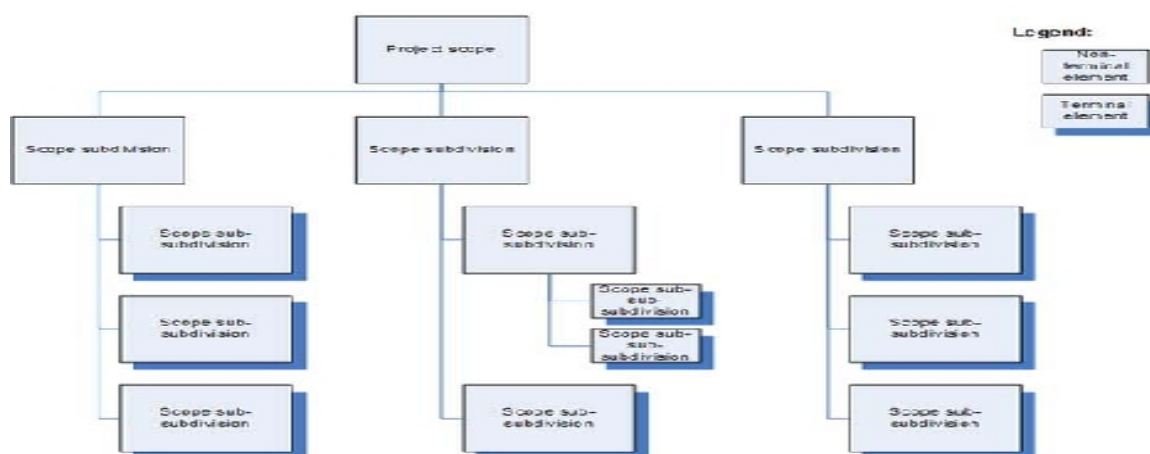


Fig. 2.3 Ejemplo simple de un EDT

Fuente: http://es.wikipedia.org/wiki/Estructura_de_descomposici%C3%B3n_del_trabajo

2.6 RUP(Rational Unified Process)

2.6.1 ¿QUE ES RUP?

RUP es un proceso de desarrollo de software cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Cubre el ciclo de vida de desarrollo de software.

RUP toma en cuenta las mejores prácticas en el modelo de desarrollo de software en particular las siguientes:

- Desarrollo de software en forma iterativa (repite una acción).
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente (Modela con Unified Modeling Language,UML)
- Verifica la calidad del software.
- Controla los cambios.

RUP permite transformar los requerimientos del usuario en un sistema.

2.6.2 FASES DE RUP

2.6.2.1 Inicio

Está principalmente dirigida al entendimiento de los requerimientos y determinar el alcance del esfuerzo de desarrollo. Se define la idea, la visión y el alcance del proyecto. Esta fase incluye la parte inicial del análisis y diseño.

2.6.2.1.1 Entregables

- Un documento con la visión del proyecto.
- Un plan del proyecto que muestre las fases y las iteraciones.
- Un caso de negocio inicial el cual incluye: contexto del negocio, criterios de éxito y planificación financiera.
- El modelo de casos de uso con una lista de todos los casos de uso y los actores que puedan ser identificados.
- Un glosario inicial del proyecto.
- Un estudio inicial de riesgos.
- Una lista de los requerimientos y restricciones principales del sistema a desarrollar.
- Estándares para el prototipo inicial.
- Un mapa de navegación inicial.

- Una lista inicial de riesgos y su evaluación.
- Una lista de requisitos funcionales y no funcionales.

2.6.2.2 Elaboración

Planificar las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura del software. Esta fase culmina con la Arquitectura del ciclo de vida.

2.6.2.2.1 Entregables

- Actualización del plan de iteración.
- Generar una lista revisada de riesgos.
- La arquitectura del software.
- Requerimientos complementarios.
- Prototipo de interfaz del usuario realizado.
- Plan de proyecto actualizado y elaborar el plan de iteración.

2.6.2.3 Construcción

Desarrollar el producto y evolucionar la visión; la arquitectura y los planes hasta que el producto en una primera versión esté listo para ser enviado a la comunidad de usuarios. Esta fase culmina con la capacidad inicial de operación.

2.6.2.3.1 Entregables

- Plan de iteración actualizado.
- Lista de riesgos revisados.
- Prototipo funcional.
- Prueba de los componentes contra los criterios de evaluación definidos.
- Plan de proyecto actualizado.

2.6.2.4 Transición

Realizar la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto hasta que el cliente esté satisfecho. Esta fase culmina con la versión definitiva de producto resultante del proyecto, la cual a su vez concluye el ciclo.

2.6.2.4.1 Entregables

- Realizar la evaluación del usuario.
- Realizar los ajustes necesarios.
- Informe de gastos.

2.6.3 ESTRUCTURA

El proceso puede describirse en dos dimensiones, a lo largo de dos ejes:

- **El eje horizontal**, representa el tiempo y muestra el aspecto dinámico del proceso, expresado en términos de ciclos, fases, iteraciones y metas.
- **El eje vertical**, representa el aspecto estático de proceso; como está descrito en términos de actividades, artefactos, trabajadores y flujos de trabajo.

Mediante la siguiente figura se puede tener una idea más clara:

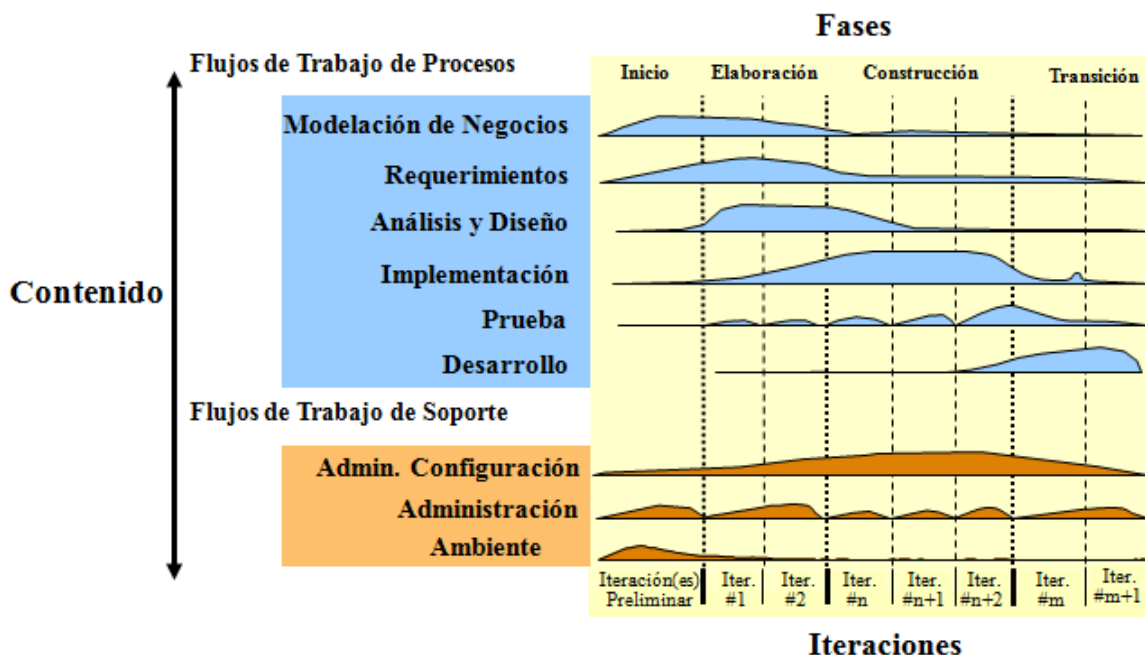


Fig. 2.4 Fases de RUP

Fuente: <https://www.u-cursos.cl/ingenieria>

2.7 ERS (Especificación de Requerimientos del Software)

2.7.1 ¿PARA QUÉ SIRVE UN ERS?

Un ERS nos permite comunicar de manera precisa los requerimientos, objetivos y presunciones del dominio y sistema a realizar. Ya que mediante este comprobamos la factibilidad del proyecto a desarrollar, esto se hace tomando nota de todas las necesidades que tiene el cliente y así sean plasmadas en el sistema final.

Se puede establecer mediante un contrato, sea este legal, documento interno o a modo de memorando. Esto nos permite tener una base para estimar (tamaño, costo, tiempo) y planificación de proyecto además es la base para la evaluación del producto final.

Mediante el contrato se puede verificar y validar el producto final, es de vital importancia tener suficiente información para decidir si el proyecto es aceptable y

satisface los requerimientos. Además nos permite tener fundamentos válidos para realizar cambios futuros y poder controlarlos.

2.7.2 CLASIFICACIÓN

A menudo los requerimientos de sistemas de software se clasifican en funcionales y no funcionales, o como requerimientos del dominio.

2.7.2.1 Requerimientos funcionales

Son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer.

Los requerimientos funcionales de un sistema describen la funcionalidad o los servicios que se espera que éste provea. Estos dependen del tipo de software y del sistema que se desarrolle y de los posibles usuarios del software. Cuando se expresan como requerimientos del usuario, habitualmente se describen de forma general mientras que los requerimientos funcionales del sistema describen con detalle la función de éste, sus entradas y salidas, excepciones, etc.

Muchos de los problemas de la ingeniería de software provienen de la imprecisión en la especificación de requerimientos. Para un desarrollador de sistemas es natural dar interpretaciones de un requerimiento ambiguo con el fin de simplificar su implementación. Sin embargo, a menudo no es lo que el cliente desea. Se tienen que estipular nuevos requerimientos y se deben hacer cambios al sistema, retrasando la entrega de éste e incrementando el costo.

En principio, la especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente. La compleción significa que todos los servicios solicitados por el usuario están definidos. La consistencia significa que los

requerimientos no tienen definiciones contradictorias. En la práctica, para sistemas grandes y complejos, es imposible cumplir los requerimientos de consistencia y compleción. La razón de esto se debe parcialmente a la complejidad inherente del sistema y parcialmente a que los diferentes puntos de vista tienen necesidades inconsistentes. Estas inconsistencias son obvias cuando los requerimientos se especifican por primera vez. Los problemas emergen después de un análisis profundo. Una vez que éstos se hayan descubierto en las diferentes revisiones o en las fases posteriores del ciclo de vida, se deben corregir en el documento de requerimientos.

2.7.2.2 Requerimientos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo, estándares, etc.

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

Muchos requerimientos no funcionales se refieren al sistema como un todo más que a rasgos particulares del mismo. Esto significa que a menudo son más críticos que los requerimientos funcionales particulares. Mientras que el incumplimiento de este último degradará el sistema, una falla en un requerimiento no funcional del sistema lo inutiliza.

Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a

factores externos como los reglamentos de seguridad, las políticas de privacidad, etcétera.

2.8 MODELO DE NEGOCIO

Un modelo de negocio (también llamado diseño de negocio o diseño empresarial) es el mecanismo por el cual un negocio trata de generar ingresos y beneficios. Es un resumen de cómo una compañía planifica servir a sus clientes. Implica tanto el concepto de estrategia como el de implementación. Comprende el conjunto de las siguientes cuestiones:

- Cómo seleccionará sus clientes
- Cómo define y diferencia sus ofertas de producto
- Cómo crea utilidad para sus clientes
- Cómo consigue y conserva a los clientes
- Cómo sale al mercado (estrategia de publicidad y distribución)
- Cómo define las tareas que deben llevarse a cabo
- Cómo configura sus recursos
- Cómo consigue el beneficio

2.9 DIAGRAMA DE PROCESOS

Es una representación gráfica de los pasos que se siguen en toda una secuencia de actividades, dentro de un proceso o un procedimiento, identificándolos mediante símbolos de acuerdo con su naturaleza; incluye, además, toda la información que se considera necesaria para el análisis.

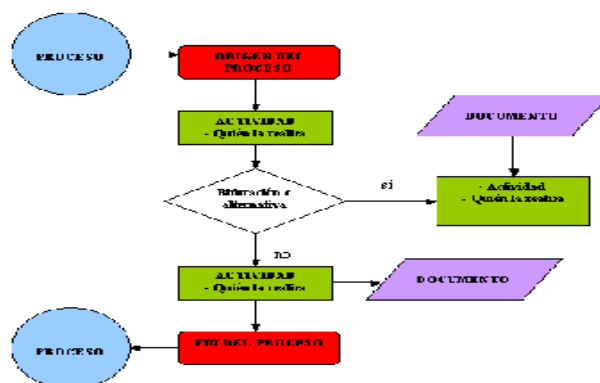


Fig. 2.5 Fases de RUP

Fuente: <http://www.gestiopolis.com/recursos/experto/catsexp/pagans/ger/36/procesos.htm>

Existen dos tipos de diagramas de procesos:

- Técnicos.
- Organizacionales.

2.9.1 DIAGRAMA DE PROCESOS TÉCNICOS

Son aquellos en donde se definen las etapas de un proceso de producción, se definen paso a paso cada una de las etapas del proceso, desde la toma de requerimientos, revisión tecnológica, generación de casos de uso, diseño de diagramas de proceso a nivel macro, diagramas de estados, modelo entidad relación, diagrama de navegación, hasta realizar la confrontación de requerimientos con el diseño inicial, para luego diseñar etapas o procedimientos adecuados.

Se afirma que un producto de calidad solo se puede conseguir cuando se dispone de procesos capaces y estables en el tiempo. El control resulta fundamental.

2.9.2 DIAGRAMA DE PROCESOS ORGANIZACIONAL

Es aquel que tiene que ver con la planeación de recurso humano y elementos organizacionales. Los pasos al desarrollar una hoja de proceso son:

- a. Hacer la hoja respectiva, cuyo encabezado tendrá datos de identificación del proceso.
- b. El cuerpo consta de 5 columnas para los símbolos anteriores, 1 para la descripción breve del trámite, 2 para las distancias de los transportes y minutos de demora y 1 para observaciones.
- c. Se anota la descripción de los pasos del proceso y se marcan puntos en las columnas de los símbolos correspondientes, uniéndolos con una línea.
- d. Se obtienen los totales, una vez terminada la descripción del proceso las operaciones, transportes, inspecciones, demoras, así como el tiempo perdido en el almacenamiento.
- e. Los totales indican el tipo de acción que conviene tomar para un análisis más profundo y cambiar aquellos aspectos que nos pueden afectar en un tiempo determinado.

2.10 BPMN (Business Process Modeling Notation)

2.10.1 ¿QUÉ ES BPMN?

Es una notación que permite estandarizar en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

2.10.2 ¿POR QUÉ ES IMPORTANTE BPMN?

Nos podemos imaginar esta situación: Estás en las primeras entrevistas de requerimientos. ¿La sorpresa? Tus usuarios no están seguros cómo funcionan sus procesos de negocio y por lo tanto resulta imposible definir los requerimientos del sistema. En este momento tienes que ponerte la camiseta de un ingeniero de negocios y quitarte la del ingeniero de software. Pero, si no cuentas con las

herramientas y técnicas adecuadas, las reuniones con tu cliente pueden ser eternas sin llegar a respuestas concretas.

2.10.3 FUNDAMENTOS DE BPMN

Un BPD (Business Process Diagram) está formado por un conjunto de elementos gráficos. Estos elementos habilitan el fácil desarrollo de diagramas simples que serán familiares para la mayoría de analistas de negocio (diagrama de flujo). Los elementos fueron elegidos para ser distinguibles los unos de los otros y para usar formas familiares para la mayoría de modeladores. Por ejemplo, las actividades son rectángulos y las decisiones son diamantes. Debe notarse que uno de los objetivos del desarrollo de BPMN es crear un mecanismo simple para crear modelos o diagramas de procesos de negocio (BPD), y al mismo tiempo que sea posible gestionar la complejidad inherente en dichos procesos. El método elegido para manejar estos dos conflictivos requisitos fue organizar los aspectos gráficos de la notación en categorías específicas. Esto da un pequeño grupo categorías que alguien que lea un BPD pueda reconocer fácilmente los tipos básicos de elementos y pueda entender el diagrama. Dentro de las categorías básicas de elementos, se puede añadir información y variaciones adicionales para dar soporte a los requerimientos complejos sin cambiar dramáticamente el *look-and-feel* básico del diagrama. Las cuatro categorías básicas de elementos son:

- Objetos de flujo
- Objetos conectores
- Artefactos
- *Swimlanes*

2.10.3.1 Objetos de flujo

Un BPD es un pequeño conjunto (tres) de elementos básicos, que son los *Objetos de Flujo*, de modo que los modeladores no tienen que aprender y reconocer un gran número de formas diferentes. Los tres objetos de flujo son:

- **Evento:** un evento se representa con un círculo. Es algo que “pasa” durante el curso del proceso de negocio. Estos eventos afectan al flujo del proceso y suelen tener una causa (trigger) o un impacto (resultado). Los eventos representados con un círculo con centro abierto permiten a los marcadores internos diferenciar diferentes triggers y resultados. Hay tres tipos de eventos, basados en cuando afectan al flujo: *Start* , *Intermediate*, y *End*.



Fig. 2.6 Objetos de flujo - Evento

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

- **Actividad:** una actividad se representa con un rectángulo redondeado y es un término genérico para el trabajo que hace una compañía. Una actividad puede ser atómica o compuesta. Los tipos que hay son: *Task* y *Sub-Process*. El Sub-Process se distingue por una pequeña marca de suma en la parte central inferior de la figura.



Fig. 2.7 Objetos de flujo - Actividad

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

- **Gateway (compuerta):** una *gateway* se representa por la típica figura de diamante y se usa para controlar la divergencia o convergencia de la secuencia de flujo. Así, esto determina las tradicionales decisiones, así como la creación de nuevos caminos, la fusión de estos o la unión. Los marcadores internos indicarán el tipo de control de comportamiento.



Fig. 2.8 Objetos de flujo - Gateway

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

2.10.3.2 Objetos conectores

Los objetos de flujo se conectan entre ellos en un diagrama para crear el esqueleto básico de la estructura de un proceso de negocio. Hay tres objetos conectores que hacen esta función. Estos conectores son:

- **Sequence Flow:** el flujo de secuencia se representa por una línea sólida con una cabeza de flecha sólida y se usa para mostrar el orden (la secuencia) en el que las diferentes actividades se ejecutarán en el Proceso. El término “control flow” normalmente no se usa en BPMN.
- **Message Flow:** el flujo de mensaje se representa por una línea discontinua con una punta de flecha hueca y se usa para mostrar el flujo de mensajes entre dos participantes del proceso separados (entidades de negocio o roles de negocio). En BPMN, dos *pools* separadas en el diagrama representan los dos participantes.
- **Association:** una asociación se representa por una línea de puntos con una punta de flecha de líneas y se usa para asociar datos, texto, y otros artefactos con los objetos de flujo. Las asociaciones se usan para mostrar entradas y salidas de las actividades.



Fig. 2.9 Objetos de flujo – Conectores

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

2.10.3.3 Swimlanes (canales)

Muchas metodologías de modelado de procesos usan el concepto de *swimlanes* como un mecanismo para organizar actividades en categorías separadas visualmente para ilustrar diferentes capacidades funcionales o responsabilidades. BPMN soporta los swimlanes con dos constructores principales. Los dos tipos de objetos swimlanes son:

- **Pool:** una *pool* representa un Participante de un Proceso. Además actúa como un contenedor gráfico para particionar un conjunto de actividades desde otros pools, normalmente en el contexto de B2B.
- **Lane:** una *lane* es una sub-partición dentro de un pool y extiende la longitud del pool, verticalmente u horizontalmente. Las lanes se usan para organizar y categorizar actividades.

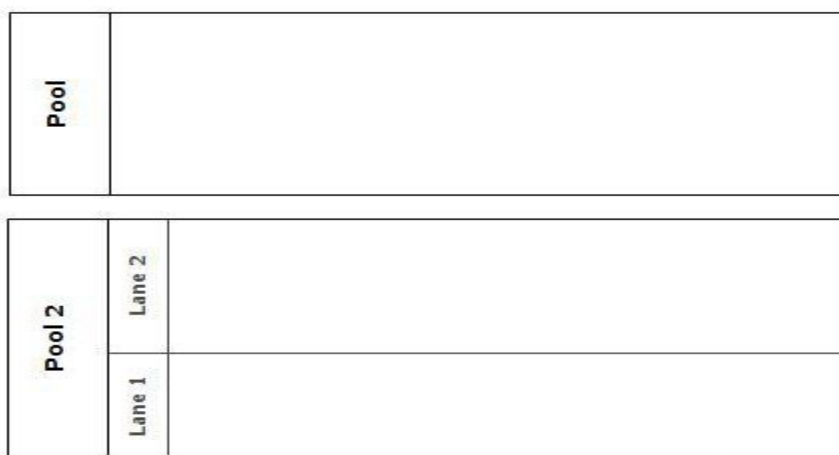


Fig. 2.10 Objetos de flujo – Canales

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

Las pools se usan cuando un diagrama implica dos entidades de negocio o participantes separados y están físicamente separados en el diagrama. Las actividades dentro de pools separadas se consideran procesos auto-contenidos. Así, el flujo de secuencia no debe cruzar el límite de un pool. El flujo de mensajes se define como el mecanismo para mostrar las comunicaciones entre dos participantes, y, de este modo debe conectar dos pools (o los objetos dentro de las pools).

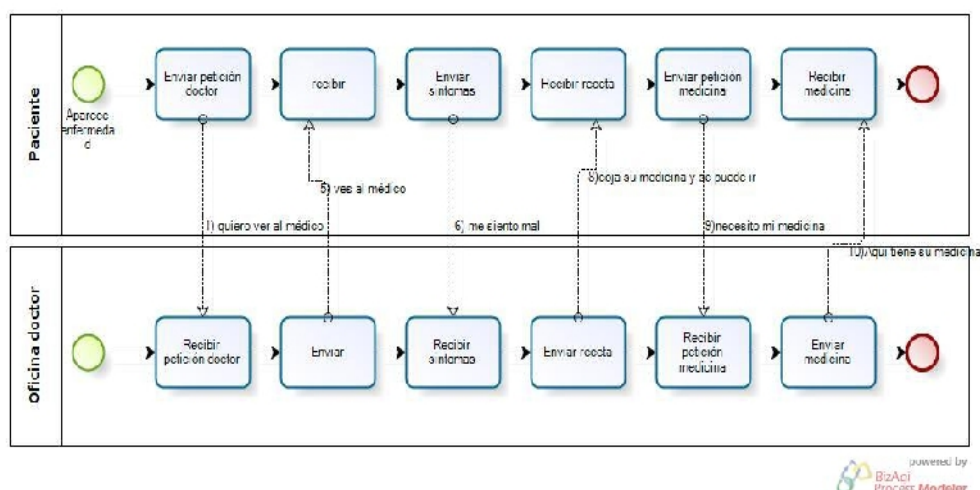


Fig. 2.11 Ejemplo de Canales

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

Las pistas (lanes) están más estrechamente relacionadas con las metodologías tradicionales de las swimlanes. Las pistas se suelen usar para separar las actividades asociadas con la función o rol de una compañía específica. El flujo de secuencia puede cruzar los límites de las pistas dentro de un pool, pero el flujo de mensajes no puede ser usado entre objetos de flujo en pistas de mismo pool.

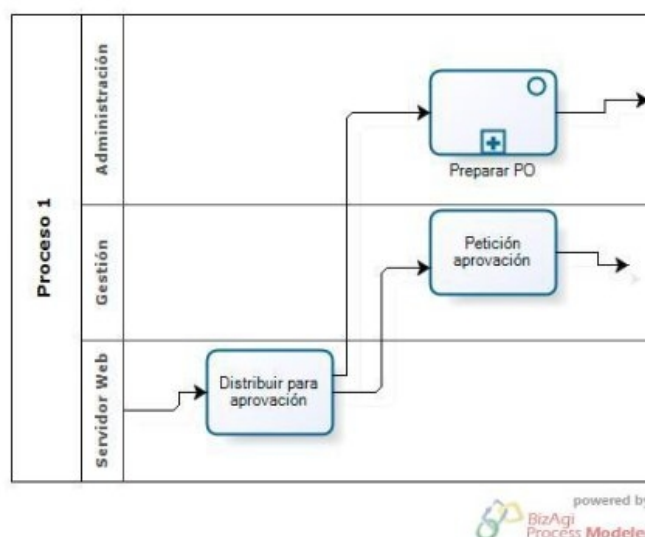


Fig. 2.12 Ejemplo de Canales

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

2.10.3.4 Artefactos

BPMN fue diseñado para permitir a los modeladores y las herramientas de modelado un poco de flexibilidad a la hora de extender la notación básica y a la hora de habilitar un contexto apropiado adicional según una situación específica, como para un mercado vertical (por ejemplo, seguros o banca). Se puede añadir cualquier número de artefactos a un diagrama como sea apropiado para un contexto de proceso de negocio específico. La versión actual de la especificación de BPMN sólo tiene tres tipos de artefactos BPD predefinidos, los cuales son:

- **Data Object:** los objetos de datos son un mecanismo para mostrar como los datos son requeridos o producidos por las actividades. Están conectados a las actividades a través de asociaciones.
- **Group:** un grupo es representado por un rectángulo redondeado con línea discontinua. El agrupamiento se puede usar documentación o análisis, pero no afecta al flujo de secuencia.
- **Annotation:** las anotaciones son mecanismos para que un modelador pueda dar información textual adicional.

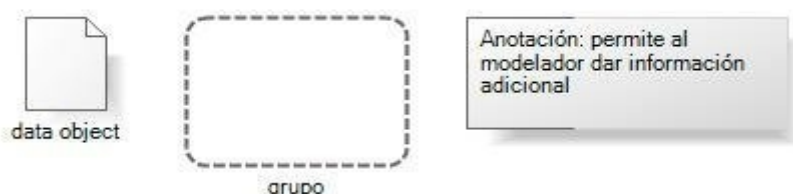


Fig. 2.13 Artefactos

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

Los modeladores pueden crear sus propios tipos de artefactos, que añaden más detalle sobre como se ejecuta el proceso – bastante a menudo para mostrar las entradas y las salidas de las actividades del Proceso. Sin embargo, la estructura básica del proceso, determinada por las actividades, gateways, y flujos de secuencia, no se cambia por añadir artefactos al diagrama.

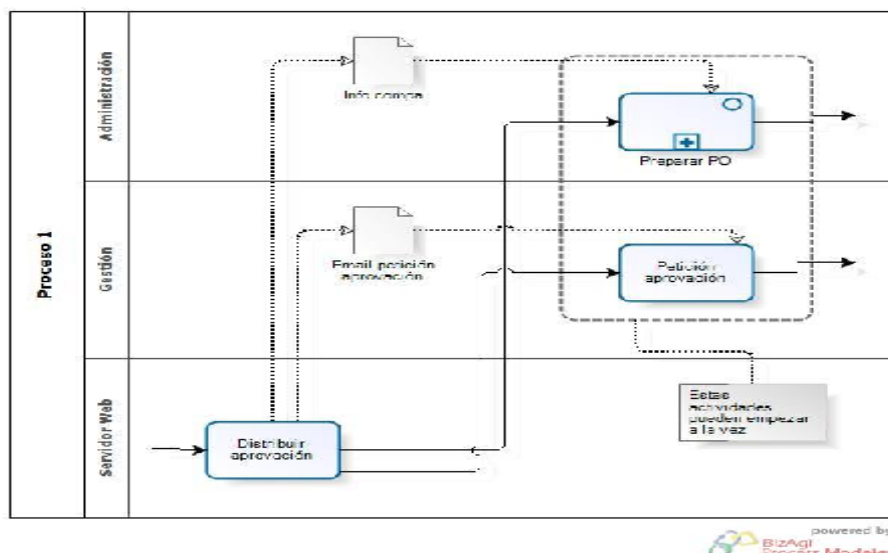


Fig. 2.14 Ejemplo de modelado de procesos

Fuente: <http://www.aprendergratis.com/introduccion-a-bpmn.html>

2.11 UML (Unified Modeling Language)

2.11.1 INTRODUCCIÓN

El Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

2.11.2 DIAGRAMAS

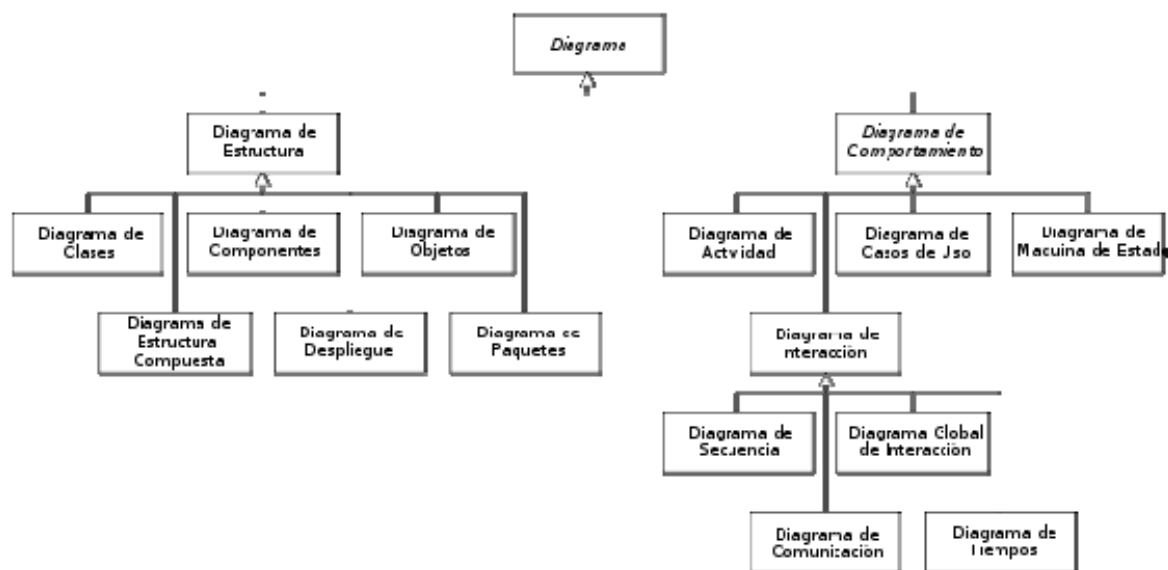


Fig. 2.15 Jerarquía de los diagramas UML 2.0, mostrados como un diagrama de clases

Fuente: http://es.wiki.org/Diagramas_UML

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la figura de la derecha.

2.11.2.1 Diagramas de Estructura

Son los que especifican como se ve el sistema modelado:

2.11.2.1.1 Diagrama de clases

Diagrama de Clases

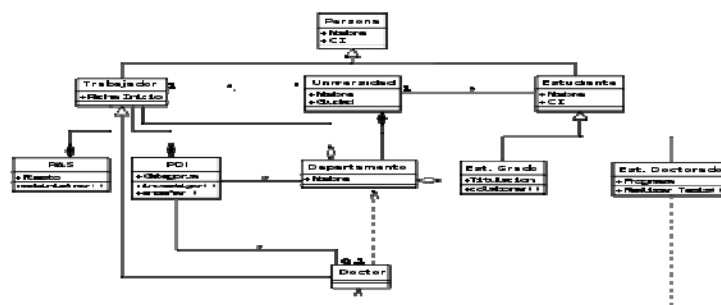


Fig. 2.16 Ejemplo de Diagrama de Clases

Fuente: http://es.wiki.org/Diagramas_UML

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

- **Propiedades**, también llamados atributos o características, son valores que corresponden a un objeto, como color, material, cantidad, ubicación. Generalmente se conoce como la información detallada del objeto. Suponiendo que el objeto es una puerta, sus propiedades serían: la marca, tamaño, color y peso.
- **Operaciones** comúnmente llamados métodos, son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar. De la misma manera que el nombre de un atributo, el nombre de una operación se escribe con minúsculas si consta de una sola palabra. Si el nombre contiene más de una palabra, cada palabra será unida a la anterior y comenzará con una letra mayúscula, a excepción de la primera palabra que comenzará en minúscula. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.
- **Interfaz** es un conjunto de operaciones que permiten a un objeto comportarse de cierta manera, por lo que define los requerimientos mínimos del objeto. Hace referencia a polimorfismo.
- **Herencia** se define como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre. Por ejemplo: Una persona puede especializarse en Proveedores, Acreedores, Clientes, Accionistas, Empleados; todos comparten datos básicos como una persona, pero además cada uno tendrá información adicional que depende del tipo de persona, como saldo del cliente, total de inversión del accionista, salario del empleado, etc.

Al diseñar una clase se debe pensar en cómo se puede identificar un objeto real, como una persona, un transporte, un documento o un paquete. Estos ejemplos de clases de objetos reales, es sobre lo que un sistema se diseña. Durante el proceso del diseño de las clases se toman las propiedades que identifican como único al objeto y otras propiedades adicionales como datos que corresponden al objeto.

2.11.2.1.2 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Debido a que estos son más parecidos a los diagramas de casos de usos estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema.

Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

2.11.2.2 Diagramas de Comportamiento

Enfatizan en lo que debe suceder en el sistema modelado:

2.11.2.2.1 Diagrama de actividades

Representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general.

El propósito del diagrama de actividad es modelar un proceso de flujo de trabajo (workflow) y/o modelar operaciones. Una Operación es un servicio proporcionado por un objeto, que está disponible a través de una interfaz. Una Interfaz es un grupo de operaciones relacionadas con la semántica.

2.11.2.2.2 Diagramas de casos de uso

Define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso. Los **diagramas de casos de uso** son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso.

2.11.2.3 Diagramas de Interacción

Son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

2.11.2.3.1 Diagrama de secuencia

tipo de mensajes se representan con flechas con la cabeza llena. Los mensajes asíncronos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia. Se representan con flechas con la cabeza abierta. También se representa la respuesta a un mensaje con una flecha discontinua.

Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria. Durante el análisis inicial, el modelador típicamente coloca el nombre 'business' de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre 'business' es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado, o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

2.11.3 CASO DE USO

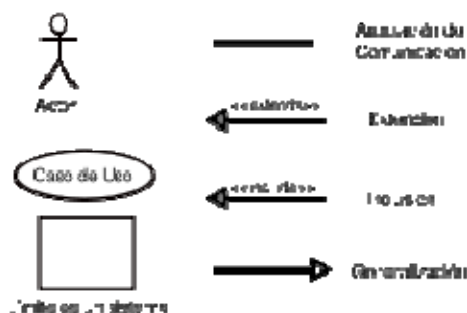


Fig. 2.18 Casos de Uso

Fuente: http://es.wiki.org/Diagramas_de_caso_de_Uso

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para

ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

2.11.3.1 Definiciones básicas

2.11.3.1.1 Actores

Se le llama Actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos así como a entidades abstractas como el tiempo.

2.11.3.2 Tipos de relaciones

2.11.3.2.1 Inclusión (include o use)

Es una forma de interacción o creación, un caso de uso dado puede "incluir" otro. El primer caso de uso a menudo depende del resultado del caso de uso incluido. Esto es útil para extraer comportamientos verdaderamente comunes desde múltiples casos de uso a una descripción individual, desde el caso de uso que lo incluye hasta el caso de uso incluido, con la etiqueta "«include»". Este uso se asemeja a una expansión de una macro, donde el comportamiento del caso incluido es colocado dentro del comportamiento del caso de uso base. No hay parámetros o valores de retorno.

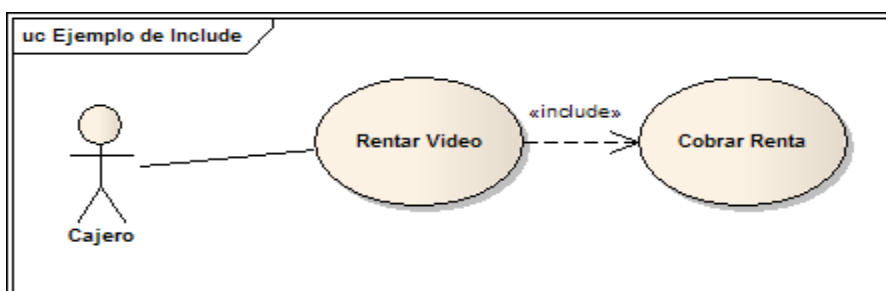


Fig. 2.19 Ejemplo Include

Fuente: http://es.wiki.org/Diagramas_de_caso_de_Uso

2.11.3.2 Extensión (Extend)

Es otra forma de interacción, un caso de uso dado, (la extensión) puede *extender* a otro. Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado en el caso de uso extendido bajo ciertas condiciones. La notación, es una flecha de punta abierta con línea discontinua, desde el caso de uso extensión al caso de uso extendido, con la etiqueta «extend». Esto puede ser útil para lidiar con casos especiales, o para acomodar nuevos requisitos durante el mantenimiento del sistema y su extensión. La extensión se utiliza en casos de uso, un caso de uso a otro caso siempre debe tener extensión o inclusión.

"La extensión, es el conjunto de objetos a los que se aplica un concepto. Los objetos de la extensión son los ejemplos o instancias de los conceptos."

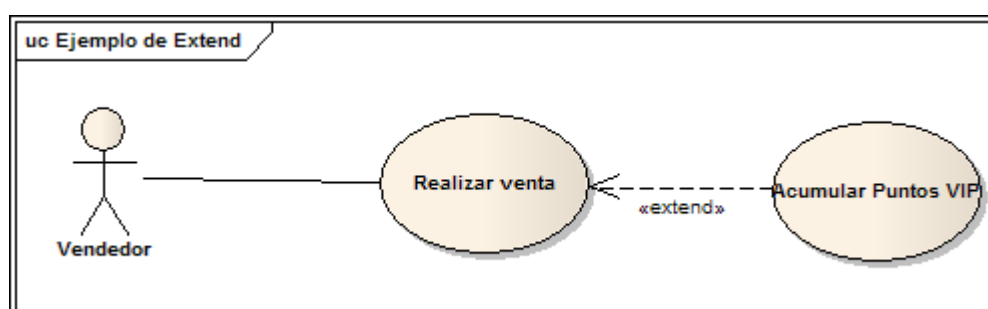


Fig. 2.20 Ejemplo Extend

Fuente: http://es.wiki.org/Diagramas_de_caso_de_Uso

Se utiliza una relación de tipo <<extends>> entre casos de uso cuando nos encontramos con un caso de uso similar a otro pero que hace algo más que éste (variante). Por contra, utilizaremos una relación tipo <<uses>> cuando nos encontramos con una parte de comportamiento similar en dos casos de uso y no queremos repetir la descripción de dicho comportamiento común.

En una relación << extends>>, un actor que lleve a cabo el caso de uso base puede realizar o no sus extensiones. Mientras, en una relación <<include>> el actor que realiza el caso de uso base también realiza el caso de uso incluido.

En general utilizaremos <<extends>> cuando se presenta una variación del comportamiento normal, y <<include>> cuando se repite un comportamiento en dos casos de uso y queremos evitar dicha repetición.

Por último en un diagrama de casos de uso, además de las relaciones entre casos de uso y actor (asociaciones) y las dependencias entre casos de uso (<<include>> y <<extends>>), pueden existir relaciones de herencia ya sea entre casos de uso o entre actores.

Llamamos modelo de casos de uso a la combinación de casos de uso y sus correspondientes diagramas. Los modelos de casos de uso se suelen acompañar por un glosario que describe la terminología utilizada. El glosario y el modelo de casos de uso son importantes puntos de partida para el desarrollo de los diagramas de clases.

Por último se debe tener en cuenta, que aunque cada caso de uso puede llevar a diferentes realizaciones, es importante reflejar en cada representación el motivo que nos ha llevado a descartarla, si es el caso.

2.11.3.3 Normas de aplicación

Los casos de uso evitan típicamente la jerga técnica, prefiriendo la lengua del usuario final o del experto del campo del saber al que se va a aplicar. Los casos de uso son a menudo elaborados en colaboración por los analistas de requerimientos y los clientes.

Cada caso de uso se centra en describir cómo alcanzar una única meta o tarea de negocio. Desde una perspectiva tradicional de la ingeniería de software, un caso de uso describe una característica del sistema. Para la mayoría de proyectos de software, esto significa que quizás a veces es necesario especificar diez o centenares de casos de uso para definir completamente el nuevo sistema. El

grado de la formalidad de un proyecto particular del software y de la etapa del proyecto influenciará el nivel del detalle requerido en cada caso de uso.

Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Un caso del uso contiene una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema. Los casos de uso no describen ninguna funcionalidad interna (oculta al exterior) del sistema, ni explican cómo se implementará. Simplemente muestran los pasos que el actor sigue para realizar una tarea.

Un caso de uso debe:

- Describir una tarea del negocio que sirva a una meta de negocio
- Tener un nivel apropiado del detalle
- Ser bastante sencillo como que un desarrollador lo elabore en un único lanzamiento

Situaciones que pueden darse:

- Un actor se comunica con un caso de uso (si se trata de un actor primario la comunicación la iniciará el actor, en cambio si es secundario, el sistema será el que inicie la comunicación).
- Un caso de uso extiende otro caso de uso.
- Un caso de uso utiliza otro caso de uso.

2.11.3.4 Ventajas

La técnica de caso de uso tiene éxito en sistemas interactivos, ya que expresa la intención que tiene el actor (su usuario) al hacer uso del sistema.

Como técnica de extracción de requerimiento permite que el analista se centre en las necesidades del usuario, qué espera éste lograr al utilizar el sistema, evitando

que la gente especializada en informática dirija la funcionalidad del nuevo sistema basándose solamente en criterios tecnológicos.

A su vez, durante la extracción (*elicitation* en inglés), el analista se concentra en las tareas centrales del usuario describiendo por lo tanto los casos de uso que mayor valor aportan al negocio. Esto facilita luego la priorización del requerimiento.

2.11.3.5 Limitaciones

Los casos de uso pueden ser útiles para establecer requisitos de comportamiento, pero no establecen completamente los requisitos funcionales ni permiten determinar los requisitos no funcionales. Los casos de uso deben complementarse con información adicional como reglas de negocio, requisitos no funcionales, diccionario de datos que complementen los requerimientos del sistema. Sin embargo la ingeniería del funcionamiento especifica que cada caso crítico del uso debe tener un requisito no funcional centrado en el funcionamiento asociado.

2.12 SERVIDOR WEB

2.12.1 ¿QUE ES UN SERVIDOR WEB?

Un servidor Web es un ordenador que usa el protocolo http para enviar páginas Web al ordenador de un usuario cuando el usuario las solicita.

Todo ordenador tiene un servidor o máquina dedicada, la cual está conectada a Internet para dar disponibilidad a sitios Web, cuando son solicitadas por usuarios de la red. Este ordenador o máquina, es conocida como servidor Web. Este equipo acepta las peticiones “HTTP⁵” del navegador Web del usuario, y entrega

⁵ Hypertext Transfer Protocol (*protocolo de transferencia de hipertexto*), ver en el glosario

las páginas Web que podemos ver, las cuales suelen ser páginas HTML⁶ y objetos enlazados, imágenes, ficheros, etc. En otras palabras, cuando alguien quiere ver una página Web en cuestión pinchando un enlace, el ordenador donde lanza una petición al servidor Web donde dicha página está alojado.

2.12.2 FUNCIONAMIENTO

El Servidor Web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador Web) y que responde a estas peticiones adecuadamente, mediante una *página Web* que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, al teclear *www.wikipedia.org* en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Además de la transferencia de código HTML, los Servidores Web pueden entregar aplicaciones Web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Comúnmente, los

⁶ Hyper Text Markup Language, (*Lenguaje de Marca de Hipertextos*), ver en el glosario

navegadores permiten ejecutar aplicaciones escritas en lenguaje *Javascript* y *java*, aunque pueden añadirse más lenguajes mediante el uso de *plugins*.

- Aplicaciones en el lado del servidor: el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador Web básico puede utilizar este tipo de aplicaciones.

El hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un lenguaje de marcas y HTTP es un protocolo.

2.13 MySQL.

MySQL es un gestor de base de datos sencillo de usar y increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que para aplicaciones no comerciales no es necesario un coste de licenciamiento.

Estas son algunas de las características principales:

- Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y

segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.

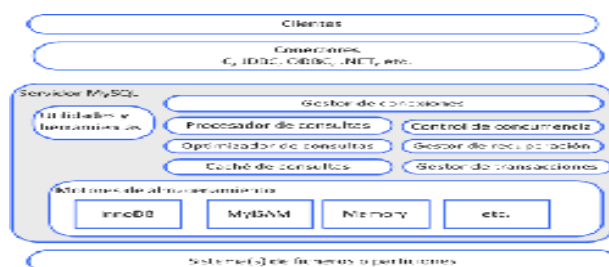
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL⁷ para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad.

2.13.1 ARQUITECTURA DE MySQL.

La arquitectura de MySQL tiene como característica más notable el separar el motor de almacenamiento (que se encarga de los detalles de entrada-salida y representación de la información en memoria secundaria) del resto de los componentes de la arquitectura. Es decir, el diseño del gestor está preparado para que se pueda cambiar el gestor de almacenamiento. Esto permite incluso crear nuevos motores de almacenamiento especializados para ciertas tareas o tipos de aplicaciones.

2.13.1.1 Arquitectura Lógica de MySQL.

La siguiente figura es una visión abstracta de la arquitectura lógica de MySQL. La figura hace una división entre los componentes que conforman el servidor, las aplicaciones cliente que lo utilizan y las partes del sistema operativo en las que se basa el almacenamiento físico.



⁷ General Public License

Fig. 2.21 Arquitectura Lógica

Autor: Miguel Ángel Sicilia

Fuente: Visión General de la Arquitectura de MySQL, Connexions

Las utilidades y herramientas de MySQL son los programas y aplicaciones que se incluyen con la distribución del gestor, o que pueden instalarse como aplicaciones adicionales. Estas incluyen las herramientas de backup, el navegador de consultas (QueryBrowser), las aplicaciones administrativas de interfaz gráfico y la herramienta de diseño MySQL Workbench, entre otras.

2.14 ASP

2.14.1 ¿QUÉ ES ASP?

El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del Web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.

Microsoft introdujo esta tecnología llamada Active Server Pages en diciembre de 1996, por lo que no es nada nueva. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente, traduciendo la definición de *Microsoft*: “Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el Web”.

El principio de la tecnología ASP es el VBScript, pero existe otra diversidad de lenguajes de programación que pueden ser utilizados como lo es Perl, JScript, etc.

2.14.2 DESCRIPCIÓN

La tecnología ASP está estrechamente relacionada con el modelo tecnológico y de negocio de su fabricante. Intenta ser solución para un modelo de programación rápida ya que "programar en ASP es como programar en Visual Basic y C#", por supuesto con muchas limitaciones y algunas ventajas específicas en entornos Web.

Lo interesante de este modelo tecnológico es poder utilizar diversos componentes ya desarrollados como algunos controles ActiveX así como componentes del lado del servidor, tales como CDONTS, por ejemplo, que permite la interacción de los scripts con el servidor SMTP que integra IIS.

Se facilita la programación de sitios Web mediante varios objetos integrados, como por ejemplo un objeto de sesión basada en cookies, que mantiene las variables mientras se pasa de página a página.

Es limitado a solo funcionar con IIS, por lo que su uso es cuestionado por la mayoría de los programadores Web quienes prefieren otros lenguajes de programación del lado del servidor como por ejemplo PHP, Perl, Java Etc.

2.14.3 VERSIONES

Ha pasado por cuatro versiones mayores:

- ASP 1.0 (distribuido con IIS 3.0)
- ASP 2.0 (distribuido con IIS 4.0)
- ASP 3.0 (distribuido con IIS 5.0)
- ASP.NET (parte de la plataforma .NET de Microsoft).

Las versiones pre-.NET se denominan actualmente (desde 2002) como ASP *clásico*.

En el último ASP clásico, ASP 3.0, hay siete objetos integrados disponibles para el programador: Application, ASPError, Request, Response, Server, Session yObjectContext. Cada objeto tiene un grupo de funcionalidades frecuentemente usadas y útiles para crear páginas Web dinámicas.

Desde 2002, el ASP *clásico* está siendo reemplazado por ASP. NET, que entre otras cosas, reemplaza los lenguajes interpretados como VBScript o JScript por lenguajes compilados a código intermedio (llamado MSIL o Microsoft Intermediate Language) como Visual Basic .NET, C#, o cualquier otro lenguaje que soporte la plataforma .NET. El código MSIL se compila con posterioridad a código nativo.

2.15 JAVASCRIPT

2.15.1 ¿QUÉ ES JAVASCRIPT?

Es un lenguaje de scripting basado en objetos sin tipo y liviano, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador Web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas Web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de *Mocha*, el cuál fue renombrado posteriormente a *LiveScript*, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador Web Netscape Navigator en la versión 2.0B3 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación Web.

Microsoft dio como nombre a su dialecto de JavaScript, JScript, para evitar problemas relacionados con la marca. JScript fue adoptado en la versión 3.0 de Internet Explorer, liberado en agosto de 1996, e incluyo compatibilidad con el Efecto 2000 Con las funciones de fecha, una diferencia de los que se basaban en ese momento. Los dialectos pueden parecer tan similares que los términos "JavaScript" y "JScript" a menudo se utilizan indistintamente, pero la especificación de JScript es incompatible con la de ECMA en muchos aspectos.

Netscape envió JavaScript a ECMA para su estandarización, resultando la versión normalizada llamada ECMAScript

2.16 ARQUITECTURA DE SOFTWARE

2.16.1 INTRODUCCIÓN

En los inicios de la informática, la programación se consideraba un arte y se desarrollaba como tal, debido a la dificultad que entrañaba para la mayoría de las personas, pero con el tiempo se han ido descubriendo y desarrollando formas y guías generales, con base a las cuales se puedan resolver los problemas. A estas, se les ha denominado Arquitectura de Software, porque, semejanza de los planos de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software. En el libro "An introduction to Software Architecture", David Garlan y Mary Shaw definen que la Arquitectura es un nivel

de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema".⁸

2.16.2 RESEÑA HISTÓRICA

En los años 1960 ya se acariciaba el concepto de arquitectura de software en los círculos de investigación (por ejemplo, por Edsger Dijkstra). No obstante, toma popularidad en los años 1990 tras reconocerse la denominada crisis del software y como tema de interés de la incipiente disciplina de la ingeniería del software.

2.16.3 ARQUITECTURA

- La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.
- Una Arquitectura de Software, también denominada *Arquitectura lógica*, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.
- La Arquitectura de Software establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades.
- Una arquitectura de software se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son

⁸ Shaw, D. G. *An Introduction to Software Architecture*.

aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.

- La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

2.16.4 ARQUITECTURAS MÁS COMUNES

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Así, las arquitecturas más universales son:

- **Monolítica.** Donde el software se estructura en grupos funcionales muy acoplados.

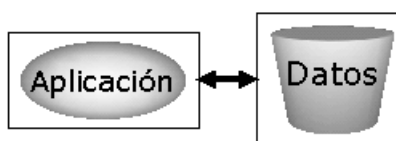


PC

Fig. 2.22 Arquitectura Monolítica: aplicación y datos en un sólo cliente aislado.

Fuente: Zapata Sánchez, Andrés Felipe. Arquitectura de Software www.fi.uba.ar

- **Cliente-servidor.** Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.



Cliente

Servidor

Fig. 2.23 Arquitectura Cliente/servidor

Fuente: Zapata Sánchez, Andrés Felipe. *Arquitectura de Software* www.fi.uba.ar

- **Arquitectura de tres niveles.** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

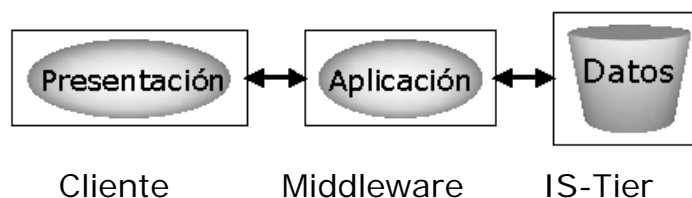


Fig. 2.24 Arquitectura Tres Capas

Fuente: Zapata Sánchez, Andrés Felipe. *Arquitectura de Software* www.fi.uba.ar

Otras arquitecturas afines menos conocidas son:

- En pipeline.
- Entre pares.
- En pizarra.
- Orientada a servicios.
- Máquinas virtuales

2.17 WAP (Wireless Application Protocol)

2.17.1 ¿QUÉ ES LA TECNOLOGÍA WAP?

WAP es el acrónimo de *Wireless Application Protocol*, que podríamos traducir como Protocolo de Aplicación Inalámbrico. La tecnología WAP es realmente un

estándar impulsado por la industria del sector de las telecomunicaciones con el objetivo de proporcionar un sistema avanzado de servicios de internet para dispositivos móviles.

El WAP está siendo fuertemente impulsado desde el WAPForum, la organización encargada de definir y desarrollar el estándar.

La tecnología tiene como premisas iniciales el uso de estándares abiertos ya existentes (como los protocolos HTTP, o el XML), la independencia de la tecnología de comunicaciones móviles sobre la que se implemente (en principio, GSM, pero en el futuro podría ser GPRS o incluso UMTS) y la independencia del terminal móvil (desde un teléfono hasta un PDA).

A partir de las premisas anteriores, el conjunto de protocolos incluidos en el estándar WAP debe adaptarse a las condiciones propias de un entorno totalmente nuevo, como es el de las comunicaciones móviles. En concreto, la red de comunicaciones presenta limitaciones de ancho de banda importantes, alta latencia y cobertura intermitente, y los terminales de acceso no tienen grandes capacidades de procesamiento, ni de memoria, además de utilizar pantallas que no permiten más de 4 o 5 líneas de texto.

Todas estas particularidades se han tenido en cuenta a la hora de diseñar los protocolos y la arquitectura del sistema WAP.

2.17.2 ARQUITECTURA WAP

La Arquitectura WAP está pensada para proporcionar un “entorno escalable y extensible para el desarrollo de aplicaciones para dispositivos de comunicación móvil”. Para ello se define una estructura en capas, en la cual cada capa es accesible por la capa superior así como por otros servicios y aplicaciones a través de un conjunto de interfaces muy bien definidos y especificados. Mediante esta figura podemos ver el esquema de capas:



Fig. 2.25 Arquitectura WAP

Fuente: <http://www.melodiasmoviles.com/documentacion/wap.php>

2.17.2.1 Capa De Aplicación (Wae)

WAE significa Wireless Application Environment y es una capa de propósito general que aúna la tecnología de Internet y la tecnología de la telefonía móvil digital. El principal objetivo de esta capa de aplicación es proporcionar la interoperabilidad necesaria tanto a los fabricantes de telefonía como a los operadores que proporcionan servicios de Internet. También se encarga del desarrollo de aplicaciones para el usuario final que le permitan consultar cualquier tipo de información de Internet de una forma sencilla y además la aplicación debe correr sobre esta capa.

Para poder realizar esto, en la capa WAE se emplea el lenguaje WML (Wireless Markup Language) que es un lenguaje de marcado de texto y que es bastante similar al HTML. También se puede emplear el WMLScript de forma similar a como se emplea JavaScript y aplicaciones de telefonía móvil como pueden ser el servicio de calendario, guía telefónica, inclusión de imágenes, etc.

2.17.2.2 Capa De Sesión (Wsp)

WSP significa Wireless Session Protocol y se encarga de proporcionar un interfaz para proporcionar servicios a la capa de aplicación WAE. La capa WAE está sobre la capa de transacción de la cual adquirirá servicios. En esta capa WSP se proporcionan los servicios a los navegadores WAP mediante el uso de la codificación HTTP pero realizándolo de forma que esta codificación se transmita a través de las ondas por el aire. Proporciona también la negociación de protocolos para permitir la comunicación y permite suspender y migrar sesiones.

Los protocolos de la capa WSP están optimizados para redes que tengan un escaso ancho de banda y con una latencia relativamente grande. Los navegadores que soportan estos tipos de protocolos se denominan WSP/B y están diseñados para permitir que el proxy WAP pueda conectar el cliente WSP/B a un servidor HTTP.

2.17.2.3 Capa De Transacción (Wtp)

WTP significa Wireless Transaction Protocol y se encarga de proporcionar servicios a los pequeños usuarios de móviles que realizan pequeñas transacciones desde estaciones móviles. La capa WTP permite operar con redes móviles seguras y no seguras basadas en datagramas y proporciona tres tipos de servicios de transacción. El primero de estos servicios es el de realizar una petición de información en un único sentido y de modo que sea no fiable. El otro servicio disponible es realizar una petición en un único sentido, pero esta vez que pueda ser fiable. Por último, el tercer servicio permite una comunicación full-dúplex, es decir, permite realizar una comunicación tanto en un sentido como en otro proporcionando además un servicio fiable.

También se puede especificar opcionalmente el envío de ACK's de reconocimiento cuando la información se transmite entre usuarios finales. Puede realizar concatenaciones de los paquetes de información o incluso retardar los ACK's para disminuir el número de mensajes enviados. También se permite las transacciones asíncronas.

2.17.2.4 Capa de seguridad (wtls) y protocolo ssl y set.

WTLS significa Wireless Transport Layer Security y es un protocolo que implementa seguridad de acuerdo al estándar de la industria SSL que significa Secure Sockets Layer y que detallaré al final de la descripción de esta capa. WTLS ha sido diseñado para su uso por parte de los protocolos de la capa de

transporte y ha sido optimizado para su uso en canales de comunicación con banda estrecha.

También ofrece la autenticación para poder autenticar el terminal y el servidor que establecen la comunicación. Por último, también ofrece la denegación del servicio para detectar y rechazar datos que son reenviados y que no son convenientemente verificados. En esta capa también se proporciona seguridad en las transacciones electrónicas económicas.

2.17.2.5 Capa De Transporte (Wdp)

WDP significa Wireless Datagram Protocol y es la capa de transporte en la arquitectura WAP. La capa de transporte WDP opera sobre las portadoras de datos que se emplean en las comunicaciones. Ofrece un servicio general de transporte a las capas superiores de la arquitectura WAP y que ya hemos mencionado anteriormente para que puedan comunicarse de forma transparente a través de la señal portadora.

2.17.3 FUNCIONAMIENTO

La tecnología WAP se basa en 3 elementos:

- **Cliente:** el cliente será el micro-navegador WML de un terminal móvil. Este será el equivalente al navegador de un PC en el internet "fijo" que todos conocemos.
- **Pasarela:** la pasarela constituye la interfaz entre la red inalámbrica y la red física. Es lo que se conoce como el *WAP Gateway*.
- **Servidor Web:** en toda arquitectura cliente / servidor tenemos un servidor encargado de procesar las peticiones del cliente y enviar las páginas solicitadas.

La siguiente figura esquematiza la interconexión de los elementos anteriores:

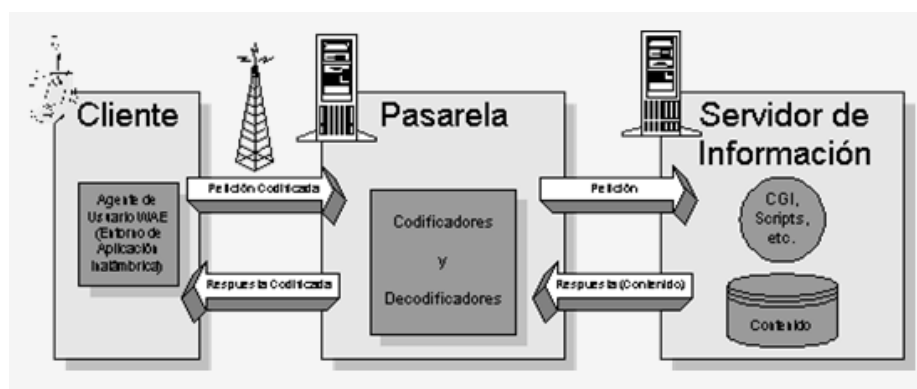


Fig. 2.26 Funcionamiento WAP

Fuente: <http://www.melodiasmoviles.com/documentacion/wap.php>

2.18 GENERACIÓN MÓVIL

2.18.1 HISTORIA DE LA TELEFONÍA CELULAR

Esta se remonta a los años 40s, cuando motivada por la segunda guerra mundial, la empresa Motorola desarrollo un teléfono móvil llamado Handie Talkie H12-16, este aparato usaba ondas de radio que para ese momento no superaban los 60 MHz.

Desde este momento, hasta principios de los años 80s diversas compañías investigaron y buscaron ofrecer un sistema de comunicación a distancia para el público general.

La historia de la telefonía celular está estrechamente ligada al desarrollo y evolución de la radio, los primeros teléfonos móviles eran demasiado grandes y pesados, por lo que su uso se restringió en principio a los autos. Generalmente el aparato se instalaba en la cabina y en el baúl se ponía un equipo de radio. El servicio no fue muy popular debido a su alto costo.

La historia y desarrollo de los teléfonos celulares se puede clasificar en etapas o generaciones:

2.18.2 PRIMERA GENERACIÓN 1G

El primer teléfono móvil fue el NMT 450 (Nordic Mobile Telephony 450 MHz), desarrollado por Ericsson, este aparato, todavía usaba las señales de radio con modulación de frecuencia.

Para esta época los teléfonos todavía eran aparatos grandes y pesados, pero constituían un gran avance en términos de comunicación móvil. Posteriormente se avanzó a frecuencias de radio (900 MHz) superiores a las ya conocidas.

Además del sistema NTM se desarrollaron simultáneamente otros sistemas como AMPS (Advanced Mobile Phone System) en EEUU y TACS (Total Access Communication System).



Fig. 2.27 Primera Generación

Fuente: http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G

2.18.3 SEGUNDA GENERACIÓN 2G

La segunda generación nace a principios de los años 90s, utiliza sistemas GSM, IS-136, iDEN e IS-95 con frecuencias de 900 y 1800 MHz.

En esta generación se abandona el uso de ondas de radio y se da paso a la era digital de las comunicaciones. Esto ofrece grandes ventajas como la calidad de voz, mejora en la seguridad y bajos costos de operación.

Se establece como estándar el GSM (Global System for Mobile communications o Groupe Spécial Mobile). Este estándar da paso con el tiempo al CDMA, que incluía nuevas ventajas sobre su predecesor. Esta etapa de transición en la telefonía celular se le conoce como generación 2.5.



Fig. 2.28 Segunda Generacion

Fuente: http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G

2.18.4 TERCERA GENERACIÓN 3G

La tercera generación se caracteriza por ampliar características como ancho de banda y manejo de datos y ofrecer nuevos servicios como la teleconferencia, televisión, acceso pleno a internet y descarga de archivos.

En este momento el desarrollo tecnológico ya posibilita un sistema totalmente nuevo: UMTS (Universal Mobile Telecommunications System). Actualmente el sistema sigue siendo desarrollado, tanto en términos de tecnología y de mercadeo.



Fig. 2.29 Tercera Generacion

Fuente: http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G

2.18.5 CUARTA GENERACIÓN 4-G

La 4G está basada totalmente en IP siendo un sistema de sistemas y una red de redes, alcanzándose después de la convergencia entre las redes de cables e inalámbricas así como en ordenadores, dispositivos eléctricos y en tecnologías de la información así como con otras convergencias para proveer velocidades de acceso entre 100 Mbps en movimiento y 1 Gbps en reposo, manteniendo una calidad de servicio (QoS) de punta a punta (end-to-end) de alta seguridad para permitir ofrecer servicios de cualquier clase en cualquier momento, en cualquier lugar, con el mínimo costo posible.

El concepto de 4G englobado dentro de 'Beyond 3-G' incluye técnicas de avanzado rendimiento radio como MIMO y OFDM. Dos de los términos que definen la evolución de 3G, siguiendo la estandarización del 3GPP, serán LTE ('Long Term Evolution') para el acceso radio, y SAE ('Service Architecture Evolution') para la parte núcleo de la red. Como características principales tenemos:

- Para el acceso radio abandona el acceso tipo CDMA característico de UMTS.
- Uso de SDR (Software Defined Radios) para optimizar el acceso radio.
- La red completa prevista es todo IP.
- Las tasas de pico máximas previstas son de 100 Mbps en enlace descendente y 50 Mbps en enlace ascendente (con un ancho de banda en ambos sentidos de 20Mhz).



Fig. 2.30 Cuarta Generacion

Fuente: http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G

2.19 TECNOLOGÍAS DE COMUNICACIÓN INALÁMBRICA

2.19.1 RED INALÁMBRICA

El término red inalámbrica (*Wireless network*) en inglés es un término que se utiliza en informática para designar la conexión de nodos sin necesidad de una conexión física (cables), ésta se da por medio de ondas electromagnéticas. La transmisión y la recepción se realizan a través de puertos. Una de sus principales ventajas es notable en los costos, ya que se elimina todo el cable Ethernet y conexiones físicas entre nodos, pero también tiene una desventaja considerable ya que para este tipo de red se debe de tener una seguridad mucho más exigente y robusta para evitar a los intrusos.

2.19.1.1 Categorías

Existen dos categorías de las redes inalámbricas.

- Larga distancia: estas son utilizadas para distancias grandes como puede ser otra ciudad u otro país.
- Corta distancia: son utilizadas para un mismo edificio o en varios edificios cercanos no muy retirados.

2.19.1.2 Tipos



Fig. 2.31 Tipos de Cobertura

Fuente: http://es.wikipedia.org/wiki/Redes_Inalambricas

Según su cobertura, se pueden clasificar en diferentes tipos:

2.19.1.2.1 Wireless Personal Area Network

En este tipo de red de cobertura personal, existen tecnologías basadas en HomeRF (estándar para conectar todos los teléfonos móviles de la casa y los ordenadores mediante un aparato central); Bluetooth (protocolo que sigue la especificación IEEE 802.15.1); ZigBee (basado en la especificación IEEE 802.15.4 y utilizado en aplicaciones como la domótica, que requieren comunicaciones seguras con tasas bajas de transmisión de datos y maximización de la vida útil de sus baterías, bajo consumo); RFID (sistema remoto de almacenamiento y recuperación de datos con el propósito de transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio).

2.19.1.2.2 Wireless Local Area Network

En las redes de área local podemos encontrar tecnologías inalámbricas basadas en HiperLAN (del inglés, *High Performance Radio LAN*), un estándar del grupo ETSI, o tecnologías basadas en Wi-Fi, que siguen el estándar IEEE 802.11 con diferentes variantes.

2.19.1.2.3 Wireless Metropolitan Area Network

Para redes de área metropolitana se encuentran tecnologías basadas en WiMAX (*Worldwide Interoperability for Microwave Access*, es decir, Interoperabilidad Mundial para Acceso con Microondas), un estándar de comunicación inalámbrica basado en la norma IEEE 802.16. WiMAX es un protocolo parecido a Wi-Fi, pero con más cobertura y ancho de banda. También podemos encontrar otros sistemas de comunicación como LMDS (*Local Multipoint Distribution Service*).

2.19.1.2.4 Wireless Wide Area Network

En estas redes encontramos tecnologías como UMTS (*Universal Mobile Telecommunications System*), utilizada con los teléfonos móviles de tercera generación (3G) y sucesora de la tecnología GSM (para móviles 2G), o también la tecnología digital para móviles GPRS (*General Packet Radio Service*).

2.20 BIZAGI

2.20.1 ¿QUE ES BIZAGI?

Fue concebido para entregar resultados rápidamente. Cuenta con un completo conjunto de herramientas las cuales a través de un ambiente grafico y sin necesidad de programación, soportan en forma ágil y flexible el ciclo de vida de un proceso de negocio: modelar, automatizar, ejecutar, gestionar y mejorar.

Es una solución altamente dinámica y poderosa que soporta el mejoramiento continuo de los procesos críticos de negocio de las empresas.

Permite representar de forma esquemática todas las actividades y decisiones que se toman en el negocio. Con una interfaz que recuerda a Microsoft Office, BizAgi Process Modeler cumple con el estándar BPMN (Business Process Management Notation).

Una vez finalizada la representación del flujo de trabajo, la aplicación puede documentar los proyectos de forma automática a partir de la información que se haya incluido en los esquemas.

Con el Modelador Bizagi, se puede hacer diagramas y documentar los procesos de la manera más eficiente y buscando fomentar la colaboración en tu organización.

Se trata de una aplicación que se puede descargar gratuitamente de Internet y utilizarla en una PC o en un ordenador portátil. Su uso es bastante sencillo y que

en cuestión de unos cuantos minutos, tendremos la capacidad de empezar a definir los procesos y colaborar con las demás personas de la organización.

Tomando en cuenta que para definir los procesos, se necesita de un trabajo en equipo, donde normalmente se ven involucradas distintas áreas de una organización. Con el Modelador de Procesos BPMN Bizagi, se comparten ideas de mejoramiento con los otros miembros del equipo, así como también presentar los procesos en un formato estándar de aceptación mundial, que ha sido conocido como BPMN: Business Process Modeling Notation.

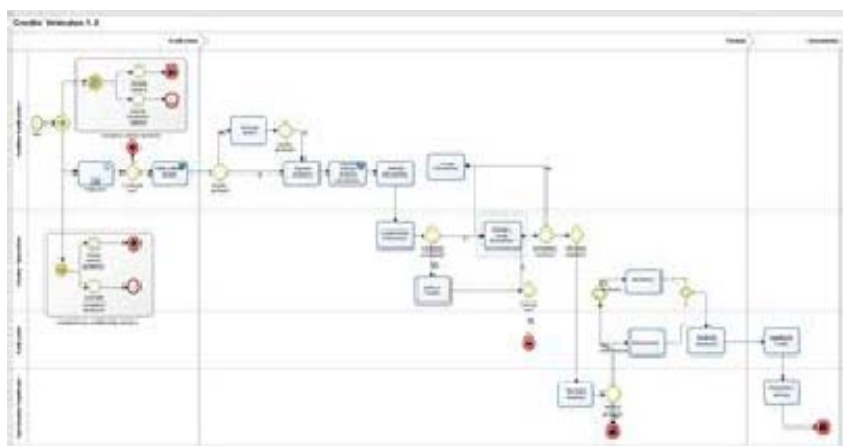


Fig. 2.32 Ejemplo Uso Bizagi

Fuente: <http://www.scribd.com/doc/22659190/Manual-De-BizAgi>

2.21 TIPOS DE PRUEBA

2.21.1 AGUANTE (STRESS TESTING)

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

2.21.2 PRUEBAS DE FUNCIONALIDAD

Una **prueba funcional** es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático.

2.21.3 PRUEBAS DE CARGA

Este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos, el servidor de aplicaciones, etc. también se monitorizan, entonces esta prueba puede mostrar el cuello de botella en la aplicación.

2.22 TIPOS DE MANUALES

2.22.1 ¿QUE ES UN MANUAL?

Un manual es el documento que contiene la descripción de actividades que deben seguirse en la realización de las funciones de una unidad administrativa, o de dos o más de ellas.

El manual incluye además los puestos o unidades administrativas que intervienen precisando su responsabilidad y participación.

Suelen contener información y ejemplos de formularios, autorizaciones o documentos necesarios, máquinas o equipo de oficina a utilizar y cualquier otro dato que pueda auxiliar al correcto desarrollo de las actividades dentro de la empresa.

En el se encuentra registrada y transmitida sin distorsión la información básica referente al funcionamiento de todas las unidades administrativas, facilita las labores de auditoria, la evaluación y control interno y su vigilancia, la conciencia en los empleados y en sus jefes de que el trabajo se está realizando o no adecuadamente.

2.22.2 MANUAL DE USUARIO

Expone los procesos que el usuario puede realizar con el sistema implantado. Para lograr esto, es necesario que se detallen todas y cada una de las características que tienen los programas y la forma de acceder e introducir información. Permite a los usuarios conocer el detalle de qué actividades ellos deberán desarrollar para la consecución de los objetivos del sistema. Reúne la información, normas y documentación necesaria para que el usuario conozca y utilice adecuadamente la aplicación desarrollada.

Al elaborar el Manual de Usuario, hay que tener en cuenta a quién va dirigido es decir, el manual puede ser manejado desde el director de la empresa hasta el introductor de datos. Por consiguiente, debe redactarse de forma clara y sencilla para que lo entienda cualquier tipo de usuario.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL PROYECTO

3.1.REQUISITOS FUNCIONALES.

Estos se encuentran dentro de la fase de Inicio, los requisitos nos permitieron tener una visión de lo que se quería tener con el sistema, gracias a ellos fue posible conocer que necesitaban en 2GETHERWEB para obtener un sistema que cumpla con las expectativas por ellos ya planteadas.

Realizado el estudio estos fueron los resultados:

3.1.1. REQUISITOS FUNCIONALES ADMINISTRADOR.

1) Ingreso con usuario y contraseña.

- El ingreso al portal móvil será con una dirección independiente, que deberá ser *mobileadmin.2gwvirtual.com*.
- Para registrar el usuario, se ingresa la dirección email de administrador.
- El idioma que se presenta para el inicio de sesión siempre será en el idioma Inglés.

2) Información de Estadísticas.

- La gráfica indica el total de afiliados por país, que se encuentran registrados en la página de 2GETHERWEB.
- El resultado que se mostrará, será apto para teléfonos móviles.

3) Respuesta de Tickets.

- El administrador observará una lista con todos los tickets recibidos.

- Además tiene la opción de responder al ticket recibido.

4) Gestión de Vector.

- El administrador visualiza las siguientes alertas.
 - Alerta de llegada de un nuevo ticket, el mismo aparecerá mientras el administrador no responda el ticket que el usuario envió.
 - Alerta de tareas pendientes por revisar en el vector interno que se maneja en la empresa.

3.1.2. REQUISITOS FUNCIONALES USUARIO.

1) Ingreso con usuario y contraseña.

- El ingreso al portal móvil será con una dirección independiente, que deberá ser *mobilen.2gwvirtual.com*.
- Para registrar el usuario, el afiliado ingresa la dirección email con la que se registró en la página.
- El idioma que se presenta para el inicio de sesión siempre será en el idioma Inglés.

2) Información de Noticias.

- Mostrar al usuario un resumen de las diez noticias más importantes, este resumen tendrá.
 - Título de la noticia.
 - Fecha de la noticia.
 - Extracto de la noticia.
 - Link para ver la noticia completa.

3) Información de la cuenta del Afiliado (Perfil).

- Presentar un resumen al afiliado, este tendrá:
 - Foto del afiliado.
 - Nombre del afiliado.

- País del afiliado.

4) Resultados de Neuroneer.

- El usuario visualiza una lista de los contactos con los que puede establecer enlaces para promocionar su proyecto.

5) Tickets

- El usuario tiene las siguientes opciones:
 - Ver todos los tickets que ha enviado mostrando la fecha de envío y de respuesta.
 - Ingreso mediante link, para ver la respuesta del ticket escogido.
 - Creación de un nuevo ticket, con las siguientes opciones:
 - Título del ticket.
 - Área que tiene el ticket.
 - Contenido del ticket.

6) Vitrina de Líderes

- El usuario observará una lista de los usuarios que tengan perfiles sobresalientes, mostrando la siguiente información:
 - Área a la que pertenece el afiliado.
 - Foto del afiliado.
 - País del afiliado.

7) Cambio de contraseña.

- El usuario tiene la opción de cambiar su contraseña, ingresando la nueva contraseña.

3.2.MODELO DE NEGOCIO.

El análisis del Modelo Negocio se encuentra en la Fase de Inicio, de esta manera se conoce como es la estructura interna de la empresa, se tiene noción de los

procesos que se desarrollan y así, saber como ir generando los módulos necesarios que componen el sistema.

3.2.1. MODELO DE NEGOCIO 2GETHERWEB COMPLETO.

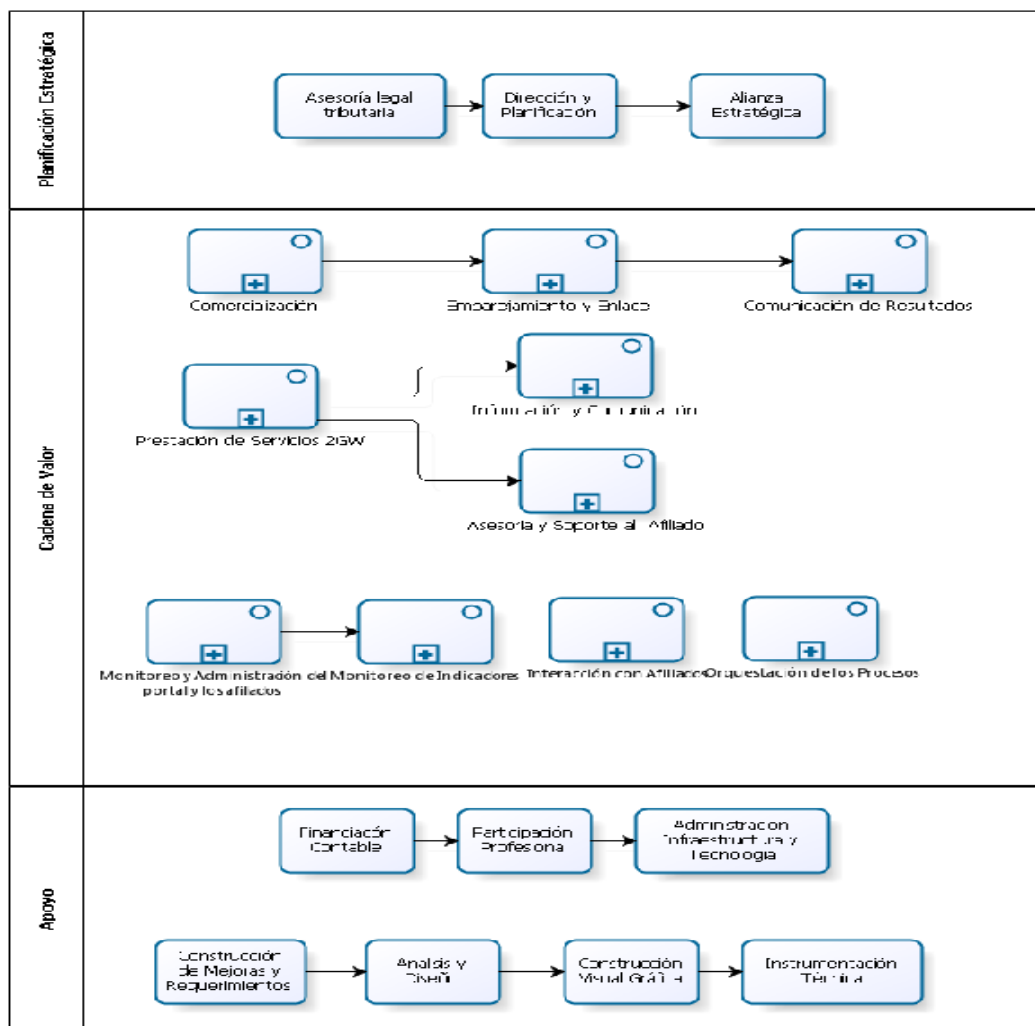


Fig.3.1 Modelo de Negocios 2GETHERWEB

Autores: Tesistas

3.2.2. CADENA DE VALOR DETALLADA.

3.2.2.1. COMERCIALIZACIÓN.

Este proceso inicia con la promoción de 2GETHERWEB, en la cual se trata de persuadir al usuario para que se afilie al portal Web, en caso de que la afiliación sea positiva se procede con la perfilación del usuario. En este punto también se promociona el portal WAP.

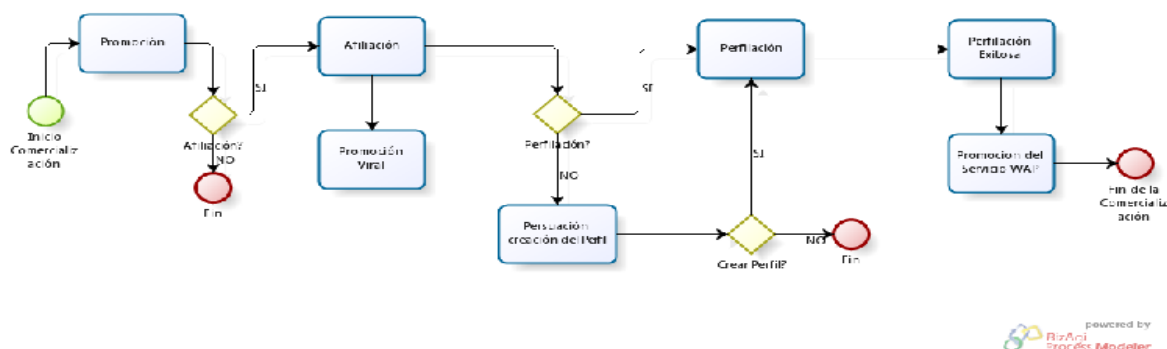


Fig.3.2 Cadena de Valor – Comercialización.

Autores: Tesistas

3.2.2.2. EMPAREJAMIENTO Y ENLACE.

Es esencial en 2GETHERWEB tener la capacidad de filtrar y buscar a los usuarios afiliados que tengan se interese con respecto a un proyecto planteado, además, se muestran los contactos más óptimos.

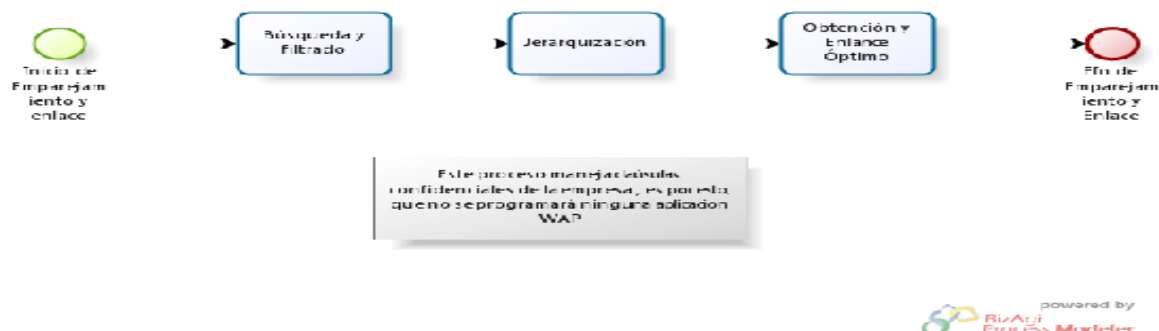
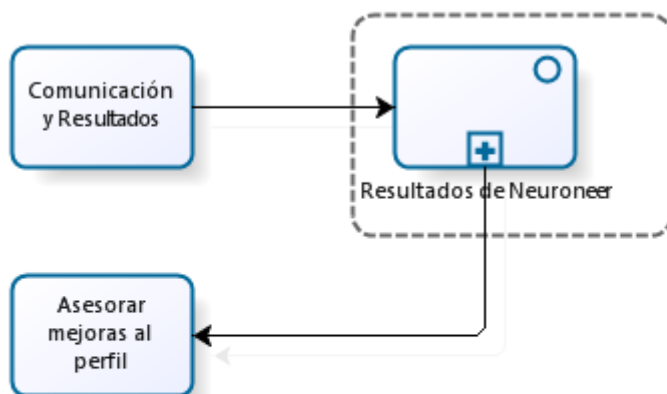


Fig.3.3 Cadena de Valor – Emparejamiento y enlace.

Autores: Tesistas

3.2.2.3. COMUNICACIÓN DE RESULTADOS

Mediante este proceso se informa de los resultados obtenidos del emparejamiento de usuarios. Visualizando los contactos, además tiene la opción de poder revisar el perfil del afiliado que propone el proyecto.

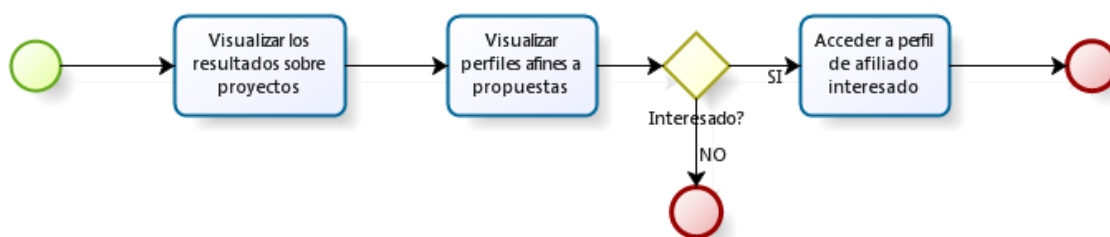


powered by
BizAgi
Process Modeler

Fig.3.4 Cadena de Valor – Comercialización de resultados.

Autores: Tesistas

3.2.2.3.1. RESULTADOS DE NEURONEER.



powered by
BizAgi
Process Modeler

Fig.3.4 Cadena de Valor – Resultados de Neuroneer.

Autores: Tesistas

3.2.2.4. PRESTACIÓN DE SERVICIOS DE 2GETHERWEB.

Es el proceso que inicia la navegación del portal Web y WAP, mediante un proceso de validación de usuarios.

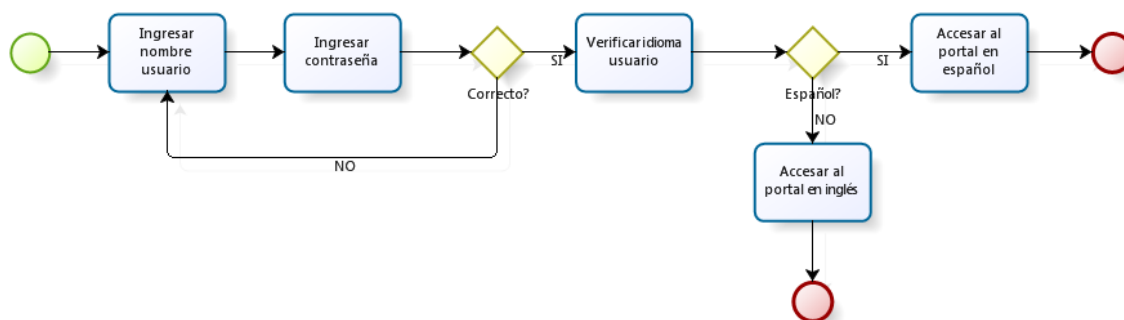


powered by
BizAgi
Process Modeler

Fig.3.5 Cadena de Valor – Prestación de servicios de 2GETHERWEB.

Autores: Tesistas

3.2.2.4.1. INICIO DE SESIÓN Y VALIDACIÓN DE USUARIOS.



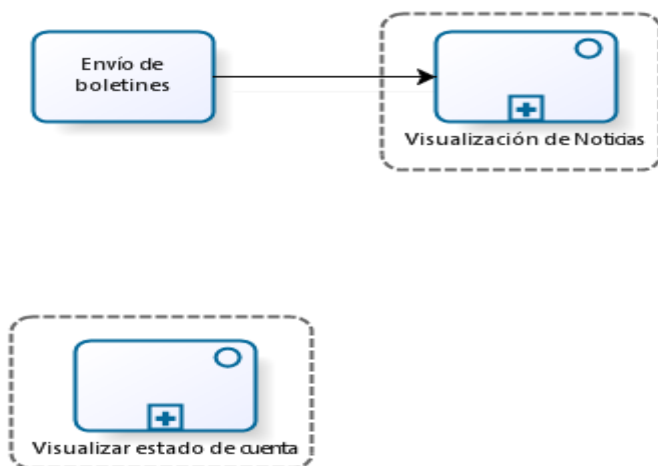
powered by
BizAgi
Process Modeler

Fig.3.6 Cadena de Valor –Inicio de Sesión y validación de usuarios.

Autores: Tesistas

3.2.2.5. INFORMACIÓN Y COMUNICACIÓN.

Se mantiene informado al usuario afiliado mediante el envío de boletines, noticias referentes a temas novedosos dentro del ámbito de negocios y su estado de cuenta.



powered by
BizAgi
Process Modeler

Fig.3.7 Cadena de Valor – Información y comunicación.

Autores: Tesistas

3.2.2.5.1. VISUALIZACIÓN DE NOTICIAS.



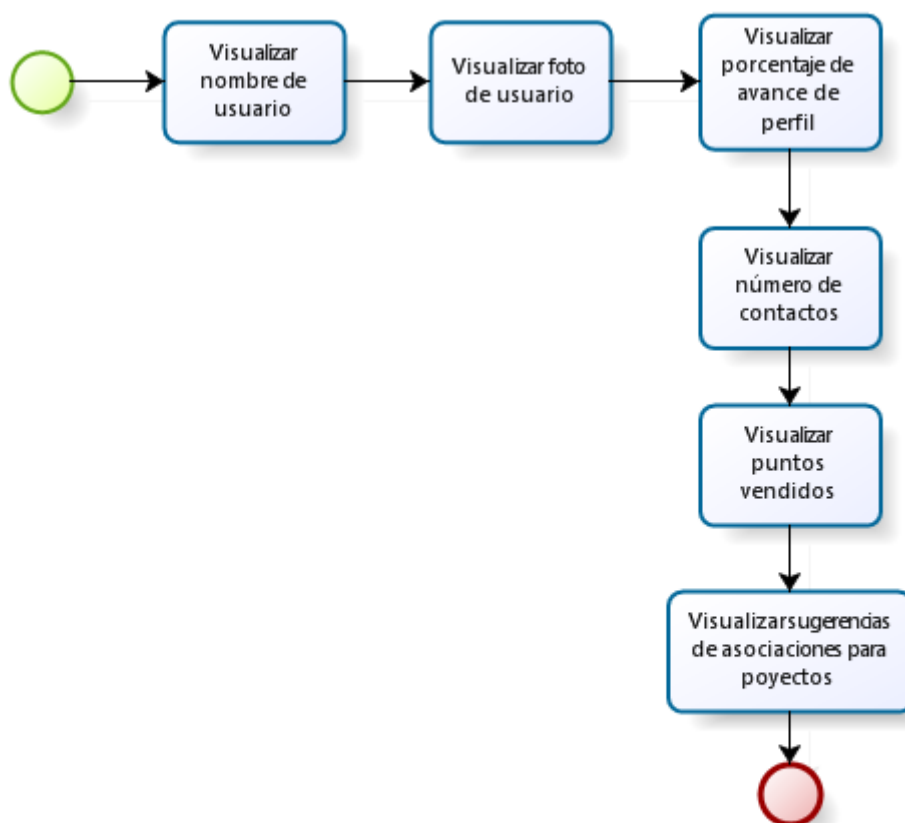
powered by
BizAgi
Process Modeler

Fig.3.8 Cadena de Valor – Visualización de noticias.

Autores: Tesistas

3.2.2.5.2. VISUALIZACIÓN DE ESTADO DE CUENTA.

En este subproceso se visualizan los principales datos del usuario afiliado, con indicadores como el numero de contactos afines a su proyecto, con esto podemos tener una idea global de la aceptación y poder de convocatoria que este tiene.



powered by
Bizagi
Process Modeler

Fig.3.9 Cadena de Valor – Visualización de estado de cuenta.

Autores: Tesistas

3.2.2.6. ASESORÍA Y SOPORTE AL AFILIADO.

El afiliado necesita asesoría permanente de las inquietudes que llegara a tener.

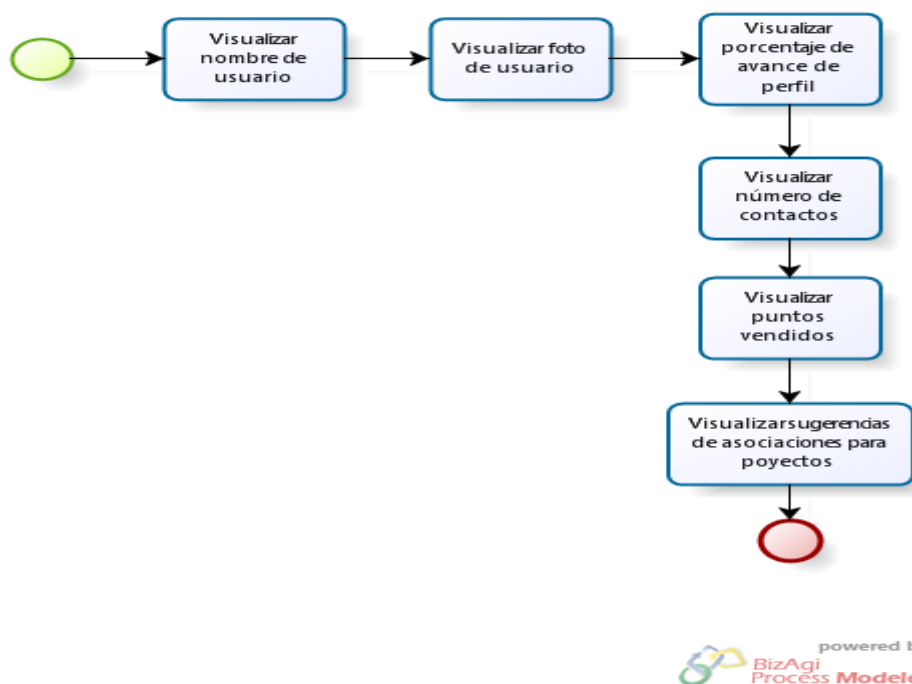


Fig.3.10 Cadena de Valor – Asesorías y soporte al afiliado.

Autores: Tesistas

3.2.2.7. MONITOREO Y ADMINISTRACIÓN DEL PORTAL Y LOS AFILIADOS.

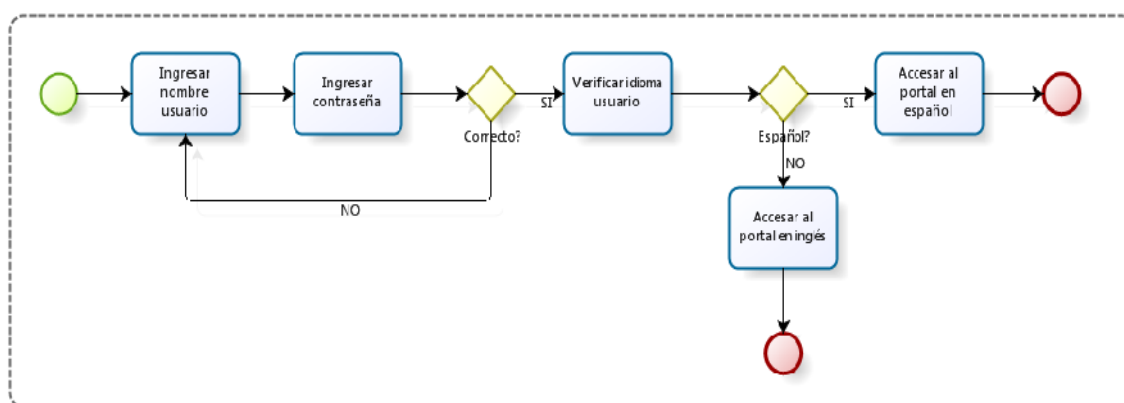


Fig.3.11 Cadena de Valor – Monitoreo y administración del portal y los afiliados.

Autores: Tesistas

3.2.2.8. MONITOREO DE INDICADORES.

El usuario administrador puede tener una visión de los principales indicadores mediante estadísticas y seguimiento de procesos o actividades.

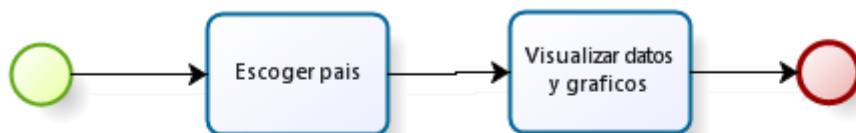


powered by
BizAgi
Process Modeler

Fig.3.12 Cadena de Valor – Monitoreo de indicadores.

Autores: Tesistas

3.2.2.8.1. ESTADÍSTICAS.



powered by
BizAgi
Process Modeler

Fig.3.13 Cadena de Valor – Estadísticas.

Autores: Tesistas

3.2.2.8.2. GESTIÓN DEL VECTOR.



Fig.3.14 Cadena de Valor – Gestión del vector.

Autores: Tesistas

3.2.2.9. INTERACCIÓN CON AFILIADOS.

El afiliado puede tener dudas acerca del negocio y su funcionamiento para lo cual existen la respuesta de tickets. Se basa en una contestación veraz d parte del usuario administrador a cualquier inquietud que tenga el cliente.



powered by
Bizagi
Process Modeler

Fig.3.15 Cadena de Valor – Interacción con afiliados.

Autores: Tesistas

3.2.2.9.1. RESPUESTA DE TICKETS.

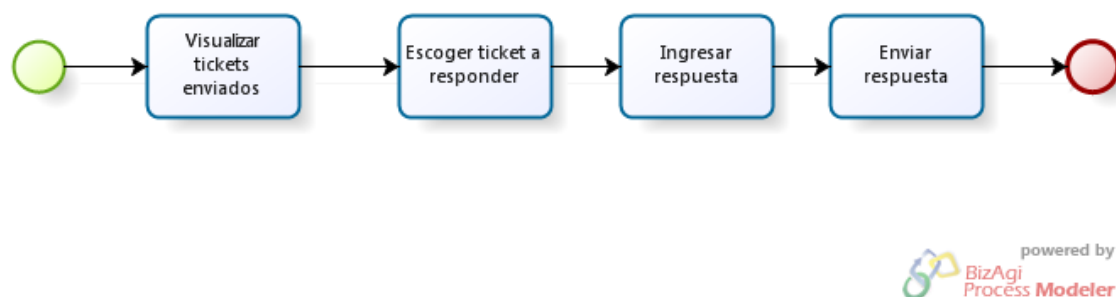


Fig.3.16 Cadena de Valor – Respuesta de tickets.

Autores: Tesistas

3.2.2.10. ORQUESTACIÓN DE PROCESOS.

De la par se monitorean actividades pendientes y avisos al usuario administrador.

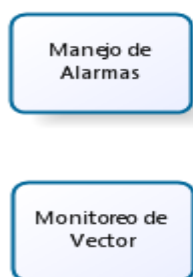


Fig.3.17 Cadena de Valor – Orquestación de procesos.

Autores: Tesistas

3.3.DIAGRAMACIÓN.

La diagramación corresponde a la fase de RUP de Análisis y Diseño, estos nos permiten tener idea de cómo desarrollar el sistema, ya que se sabrá como los distintos usuarios que existen en el sistema navegarán en el, que actividades tienen, es decir podremos conocer los accesos que tendrán en cada pantalla y como será su desenvolvimiento dentro del mismo.

3.3.1. DIAGRAMA DE CASOS DE USO.

3.3.1.1. INICIO SESIÓN USUARIO.

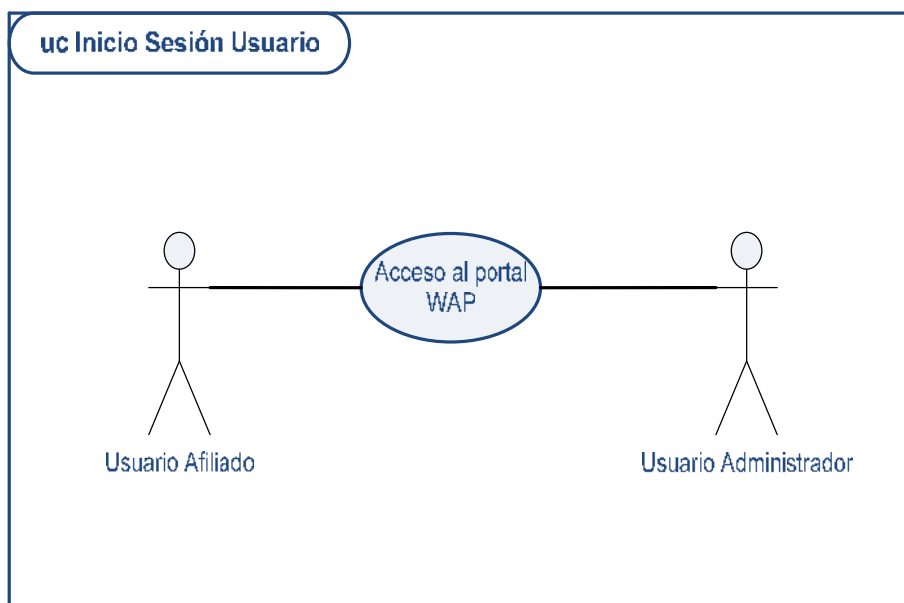


Fig.3.18 Caso de Uso – Inicio sesión usuario

Autores: Tesistas

Descripción Inicio Sesión Usuario	
Nombre:	Acceso al portal WAP.
Descripción:	Validar el acceso de los usuarios administrador y afiliado al portal WAP.
Actor:	Usuario Afiliado, Usuario Administrador.
Precondiciones:	Los usuarios deben tener una cuenta creada en 2GETHER WEB.
Garantías de éxitos:	Acceso tanto para el usuario administrador como para el usuario afiliado al portal WAP.

3.3.1.2. MENÚ AFILIADO

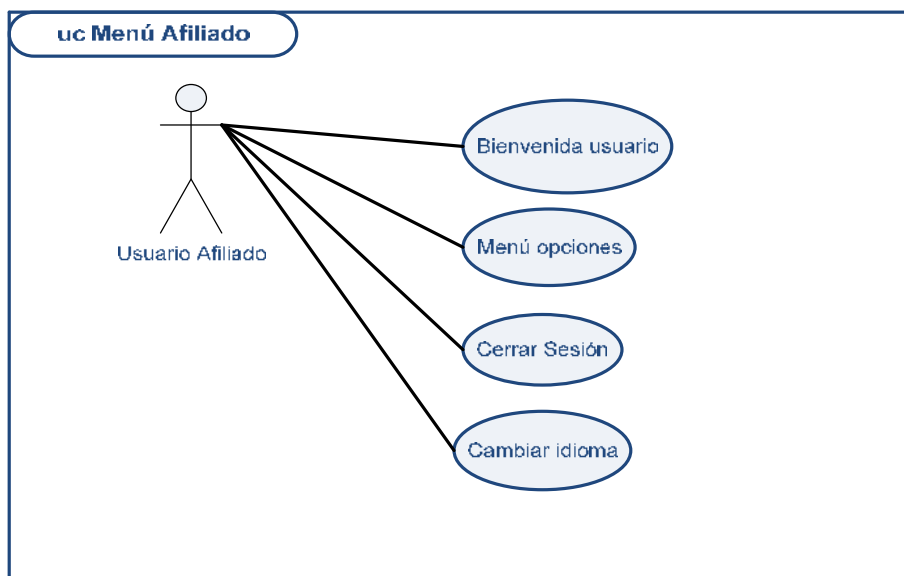


Fig.3.19 Caso de Uso – Menú Afiliado

Autores: Tesistas

Descripción Menú Afiliado	
Nombre:	Bienvenida usuario.
Descripción:	Mostrar el nombre del usuario afiliado.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe ingresar con su usuario y Password.
Garantías de éxitos:	El usuario tendrá la garantía de que ingreso en su perfil.

Descripción Menú Afiliado	
Nombre:	Menú Opciones.
Descripción:	Mostrar un menú gráfico con los accesos a las diferentes opciones del portal WAP para el usuario afiliado.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe ingresar con su usuario y Password.

Garantías de éxitos:	Navegación del usuario afiliado por el portal de acuerdo a sus necesidades.
-----------------------------	---

Descripción Menú Afiliado	
Nombre:	Cerrar Sesión.
Descripción:	Salir del portal WAP.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado desea salir del portal WAP.
Garantías de éxitos:	El usuario saldrá de manera exitosa de su perfil.

Descripción Menú Afiliado	
Nombre:	Cambiar idioma.
Descripción:	Cambiar idioma en el portal WAP.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado desea cambiar el idioma del portal WAP.
Garantías de éxitos:	El usuario obtendrá la información en inglés.

3.3.1.3. PERFIL

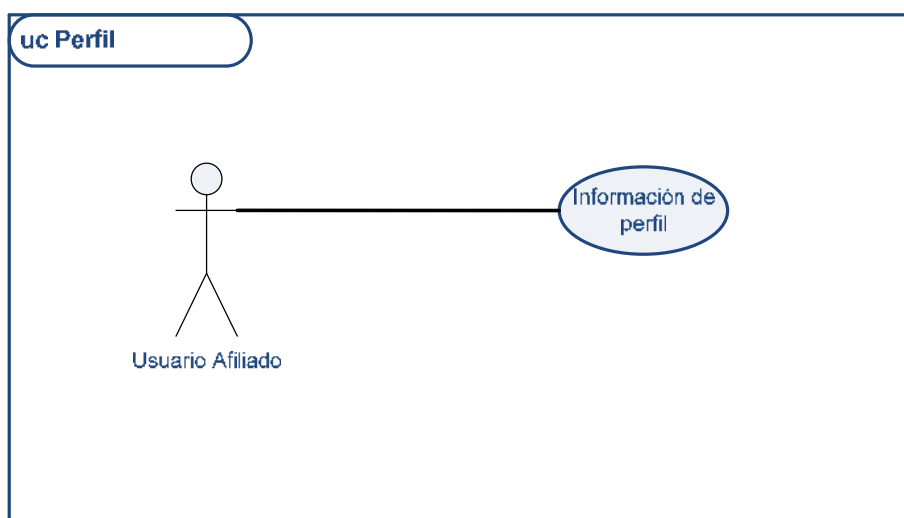


Fig.3.20 Caso de Uso –Perfil

Autores: Tesistas

Descripción Perfil	
Nombre:	Información de perfil.
Descripción:	Visualizar un resumen del perfil del usuario afiliado, indicando información sobre evolución de sus proyectos.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe tener lleno su perfil.
Garantías de éxitos:	Hacer que otros usuarios puedan acceder al perfil y conocer sobre los proyectos e intereses propuestos.

3.3.1.4. NOTICIAS RECIBIDAS

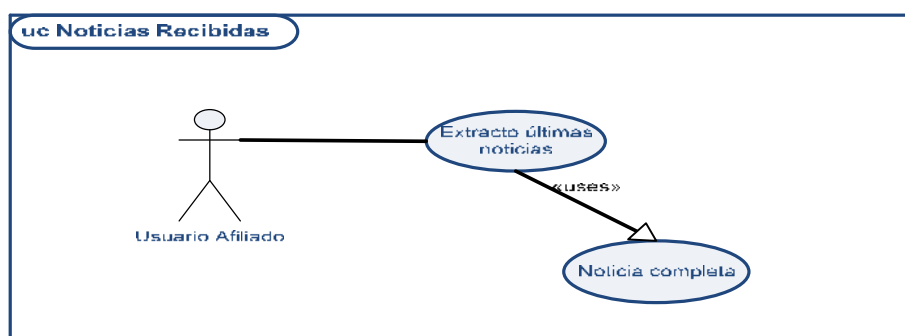


Fig.3.21 Caso de Uso – Noticias Recibidas

Autores: Tesistas

Descripción Noticias Recibidas	
Nombre:	Extracto ultimas noticias.
Descripción:	Visualizar un resumen de las últimas noticias ingresadas por fecha y título.
Actor:	Usuario Afiliado.
Precondiciones:	Las noticias deben ser ingresadas y se debe indicar un resumen concreto de las mismas.
Garantías de éxitos:	Noticias resumidas que son de interés para los afiliados que indican resultados de proyectos y avances.

Descripción Visualización Noticias	
Nombre:	Noticia completa.
Descripción:	Visualizar la noticia con toda su información.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe escoger la noticia que quiere visualizar.
Garantías de éxitos:	El usuario revisa la noticia que es de su interés.

3.3.1.5. TICKETS DE SERVICIO

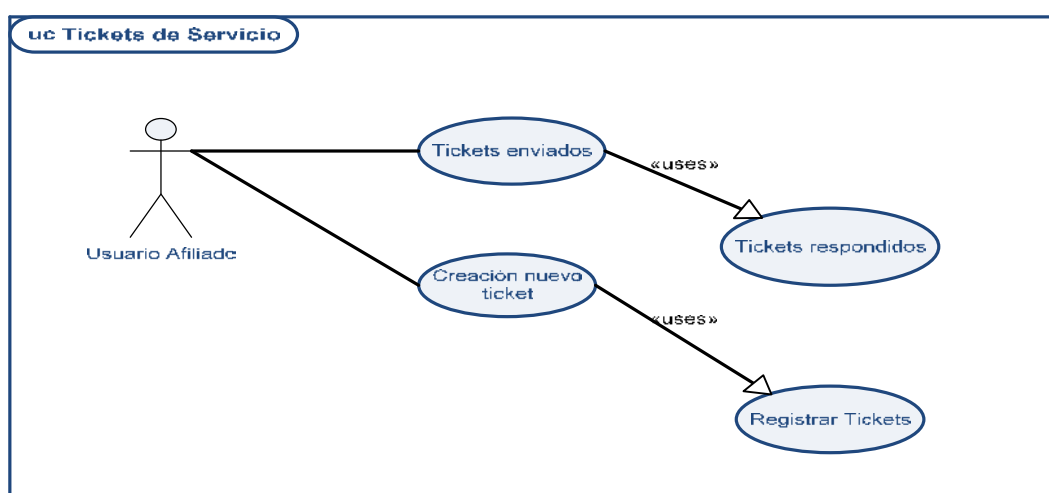


Fig.3.22 Caso de Uso – Tickets de servicio

Autores: Tesistas

Descripción Tickets de Servicio	
Nombre:	Tickets enviados.
Descripción:	El usuario afiliado podrá revisar y enviar tickets para consultar a ejecutivos de 2GETHERWEB.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe tener dudas sobre evolución o creación de sus proyectos.
Garantías de éxitos:	Envío del ticket al ejecutivo de 2GETHERWEB.

Descripción Tickets de Servicio	
Nombre:	Tickets respondidos.
Descripción:	El usuario afiliado podrá revisar que respuesta fue dada por el ejecutivo de 2GETHERWEB.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debió ingresar la duda sobre sus proyectos.
Garantías de éxitos:	Respuesta al usuario afiliado por parte del ejecutivo de 2GETHERWEB.

Descripción Tickets de Servicio	
Nombre:	Creación nuevo ticket.
Descripción:	El usuario afiliado tiene la opción de crear un nuevo ticket, el mismo que será enviado al Administrador.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe ingresar la duda sobre sus proyectos.
Garantías de éxitos:	Respuesta al usuario afiliado por parte del ejecutivo de 2GETHERWEB.

Descripción Tickets de Servicio	
Nombre:	Registrar ticket.
Descripción:	El usuario afiliado ingresa el ticket que va a enviar indicando el tema del ticket y la descripción del mismo.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe ingresar la duda sobre sus proyectos.
Garantías de éxitos:	Respuesta al usuario afiliado por parte del ejecutivo de 2GETHERWEB.

3.3.1.6. RESULTADOS NEURONEER.

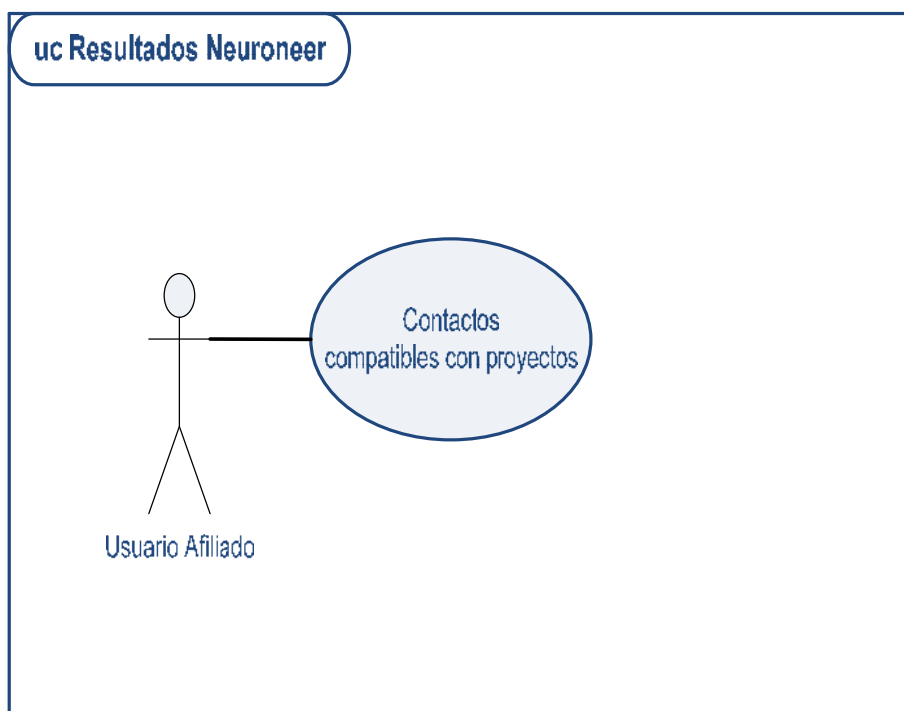


Fig.3.23 Caso de Uso – Neuroneer

Autores: Tesistas

Descripción Resultados Neuroneer	
Nombre:	Contactos compatibles con proyectos.
Descripción:	El usuario visualiza que contactos son compatibles con el proyecto que propuso y así establece el contacto con ellos.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe ingresar en su proyecto indicando en que ámbito desea desarrollarse.
Garantías de éxitos:	Conocer con que contacto puede establecer contacto.

3.3.1.7. VITRINA DE LÍDERES.

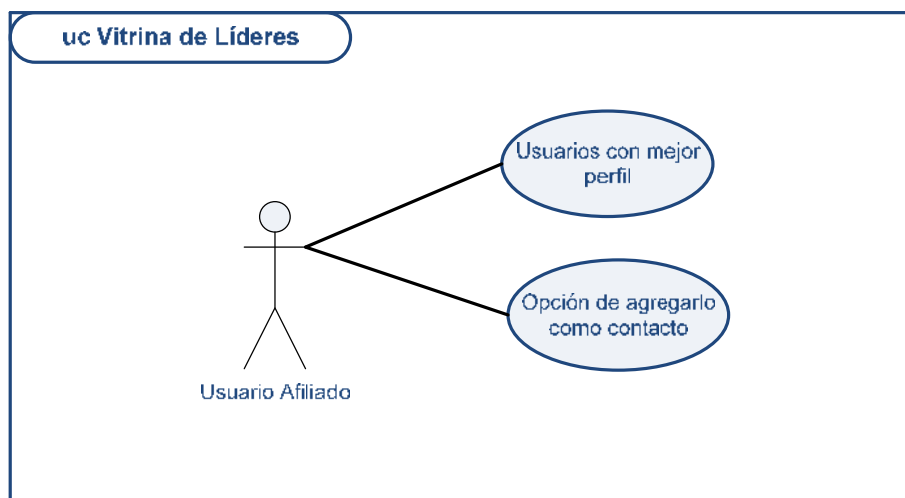


Fig.3.24 Caso de Uso – Vitrina de Líderes

Autores: Tesistas

Descripción Vitrina de Líderes	
Nombre:	Usuarios con mejor perfil.
Descripción:	El usuario visualiza los usuarios que tienen un buen perfil.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe loguearse.
Garantías de éxitos:	Conocer que afiliados pueden ser útiles para establecer contacto.

Descripción Vitrina de Líderes	
Nombre:	Opción de agregarlo como contacto.
Descripción:	El usuario puede invitar al afiliado que aparece en la vitrina para que forme parte de la lista de sus contactos.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe ingresar en la vitrina de líderes.
Garantías de éxitos:	Mandar una invitación al afiliado.

3.3.1.8. CAMBIO DE CONTRASEÑA.

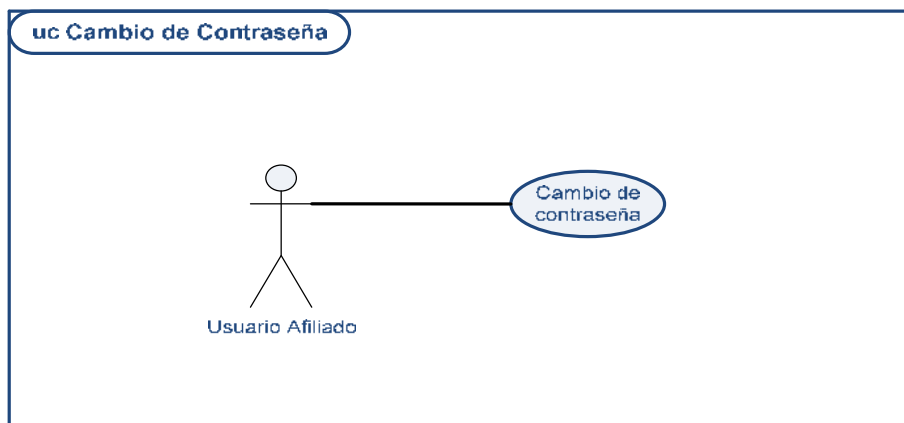


Fig.3.25 Caso de Uso – Cambio de contraseña

Autores: Tesistas

Descripción Cambio de contraseña	
Nombre:	Cambio de contraseña.
Descripción:	El usuario puede cambiar su contraseña.
Actor:	Usuario Afiliado.
Precondiciones:	El usuario afiliado debe loguearse.
Garantías de éxitos:	Contraseña cambiada.

3.3.1.9. ESTADÍSTICAS.

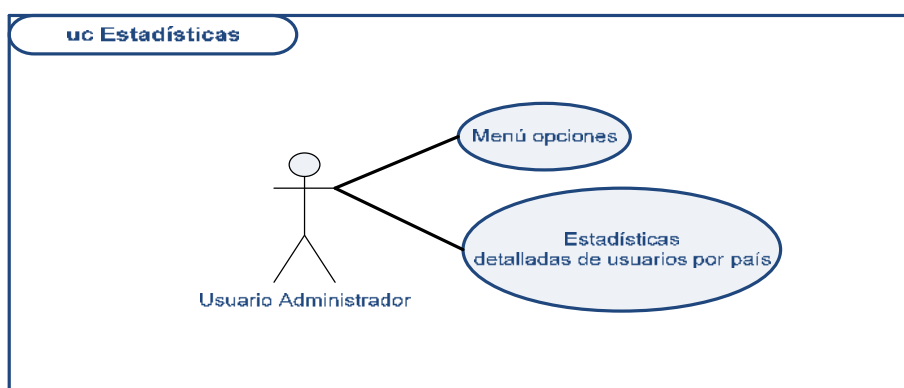


Fig.3.26 Caso de Uso – Estadísticas

Autores: Tesistas

Descripción Estadísticas	
Nombre:	Menú Opciones
Descripción:	Opciones de menú para el usuario administrador
Actor:	Usuario Administrador
Precondiciones:	El usuario administrador ingresa con su usuario y contraseña
Garantías de éxitos:	Acceso a estadísticas por parte del usuario administrador

Descripción Estadísticas	
Nombre:	Estadísticas detalladas de usuarios por país.
Descripción:	El usuario administrador revisa información de países que se encuentran afiliados a 2GETHERWEB, teniendo datos de la cantidad de afiliados que se manejan en los países donde se encuentra 2GETHERWEB.
Actor:	Usuario Administrador.
Precondiciones:	Debe existir información relacionada a los países afiliados.
Garantías de éxitos:	Grafica estadística que permita comparar el desarrollo de la empresa en varios.

3.3.1.10. RESPUESTA DE TICKETS.

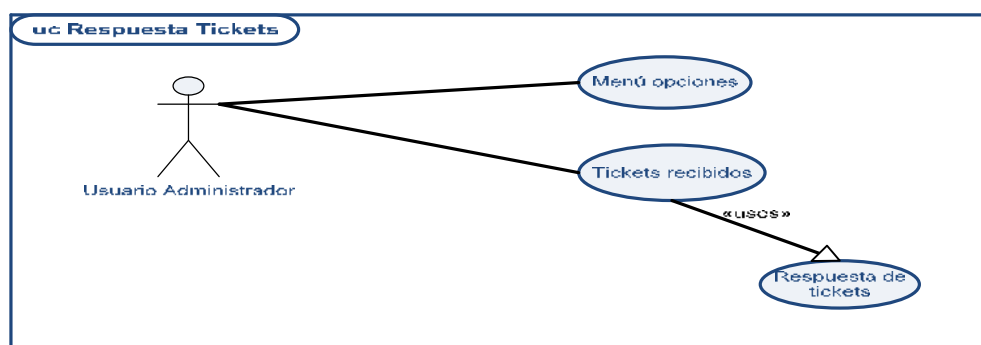


Fig.3.27 Caso de Uso – Respuesta de Tickets

Autores: Tesistas

Descripción Respuesta Tickets	
Nombre:	Menú Opciones.
Descripción:	Visualizar los tickets que ha recibido.
Actor:	Usuario Administrador.
Precondiciones:	El usuario afiliado debe mandar dudas mediante un ticket.
Garantías de éxitos:	Saber que tickets tiene que responder.

Descripción Respuesta Tickets	
Nombre:	Tickets recibidos.
Descripción:	Opciones de menú para el usuario administrador
Actor:	Usuario Administrador
Precondiciones:	El usuario administrador ingresa con su usuario y contraseña
Garantías de éxitos:	Acceso a respuesta de tickets por parte del usuario administrador

Descripción Respuesta Tickets	
Nombre:	Respuesta de tickets.
Descripción:	El usuario administrador podrá responder los tickets recibidos.
Actor:	Usuario Administrador.
Precondiciones:	El usuario administrador debe escoger el ticket que quiere responder.
Garantías de éxitos:	Mandar respuesta del ticket al usuario afiliado.

3.3.1.11. VECTOR.

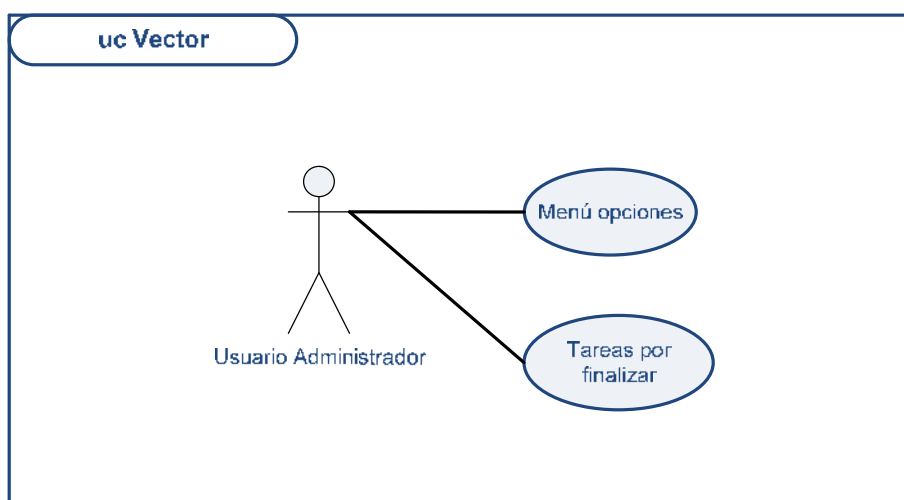


Fig.3.28 Caso de Uso – Vector

Autores: Tesistas

Descripción Respuesta Tickets	
Nombre:	Menú Opciones.
Descripción:	Visualizar los tickets que ha recibido.
Actor:	Usuario Administrador.
Precondiciones:	El usuario afiliado debe mandar dudas mediante un ticket.
Garantías de éxitos:	Saber que tickets tiene que responder.

Descripción Vector	
Nombre:	Tareas por finalizar.
Descripción:	El usuario administrador visualiza que tareas pendientes tiene que finalizarlas, además puede ver si tiene tickets pendientes por responder.
Actor:	Usuario Administrador.
Precondiciones:	El usuario administrador debe loguearse.
Garantías de éxitos:	Tareas finalizadas.

3.3.2. DIAGRAMA DE ACTIVIDADES.

3.3.2.1. INICIAR SESIÓN USUARIO.

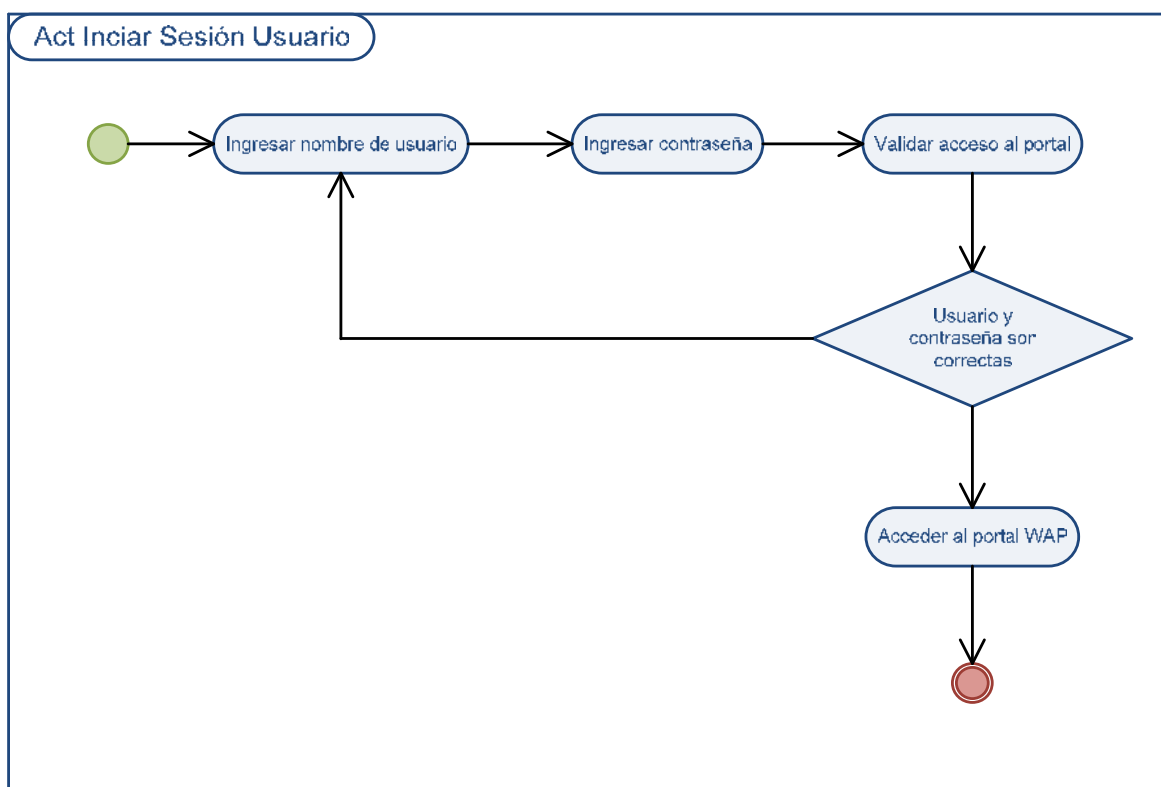


Fig.3.29 Actividades – Iniciar Sesión usuario.

Autores: Tesistas

3.3.2.2. VISUALIZAR PERFIL.

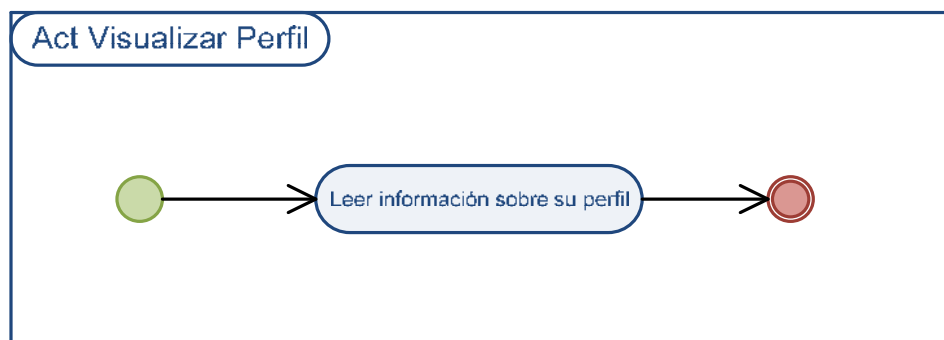


Fig.3.30 Actividades – Visualizar perfil.

Autores: Tesistas

3.3.2.3. VISUALIZAR NOTICIAS.

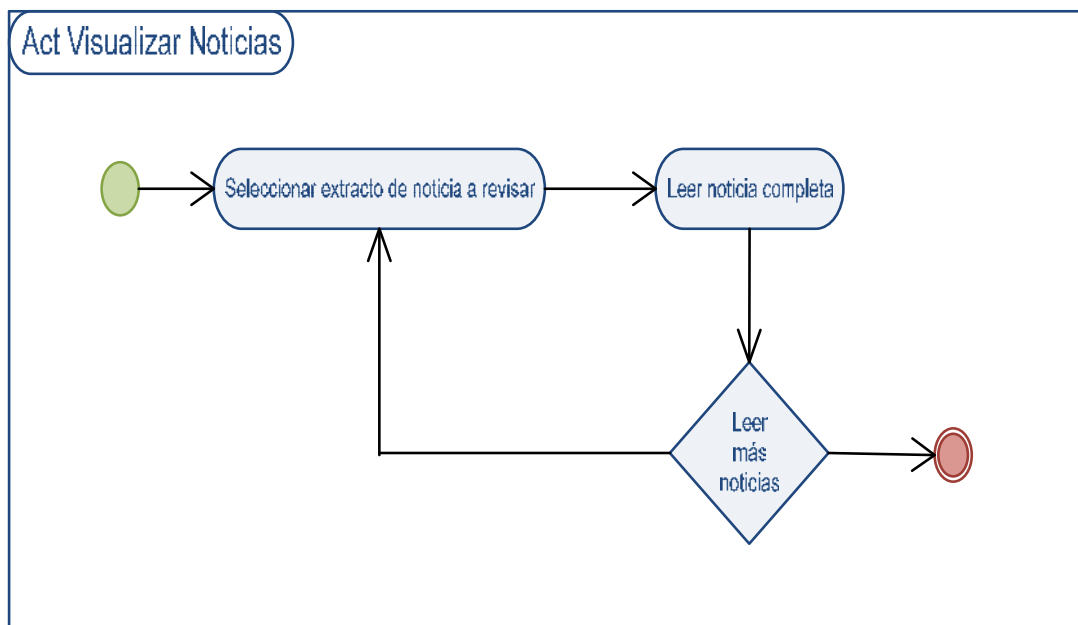


Fig.3.31 Actividades – Visualizar noticias.

Autores: Tesistas

3.3.2.4. CREAR TICKETS.

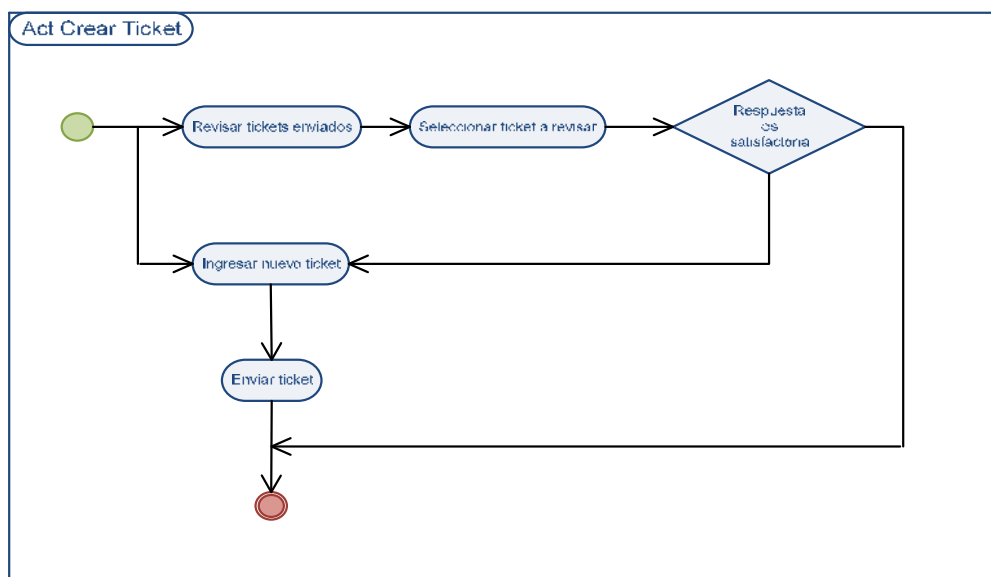


Fig.3.32 Actividades – Crear tickets.

Autores: Tesistas

3.3.2.5. VISUALIZAR NEURONEER.

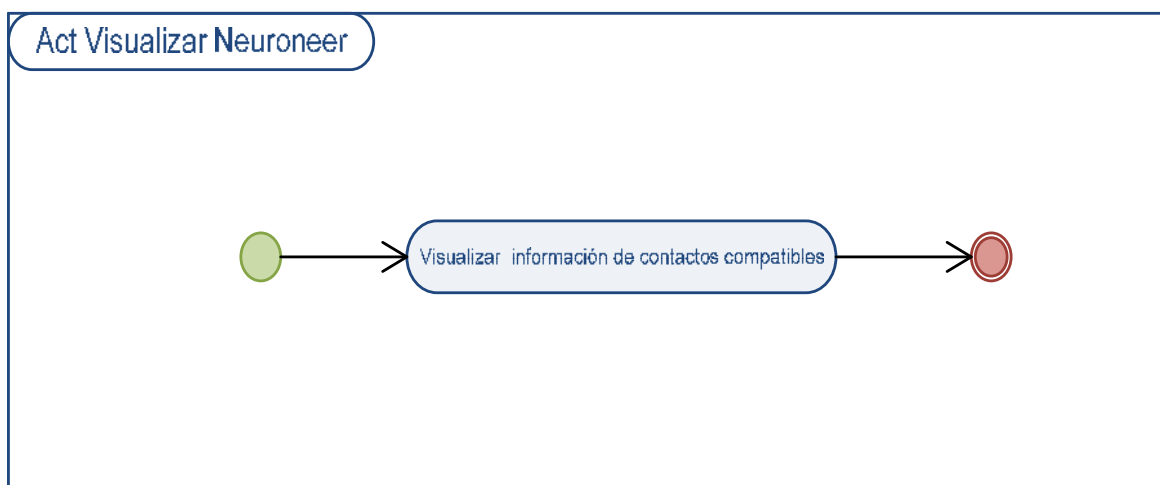


Fig.3.33 Actividades – Visualizar Neuroneer.

Autores: Tesistas

3.3.2.6. VISUALIZAR VITRINA DE LÍDERES.

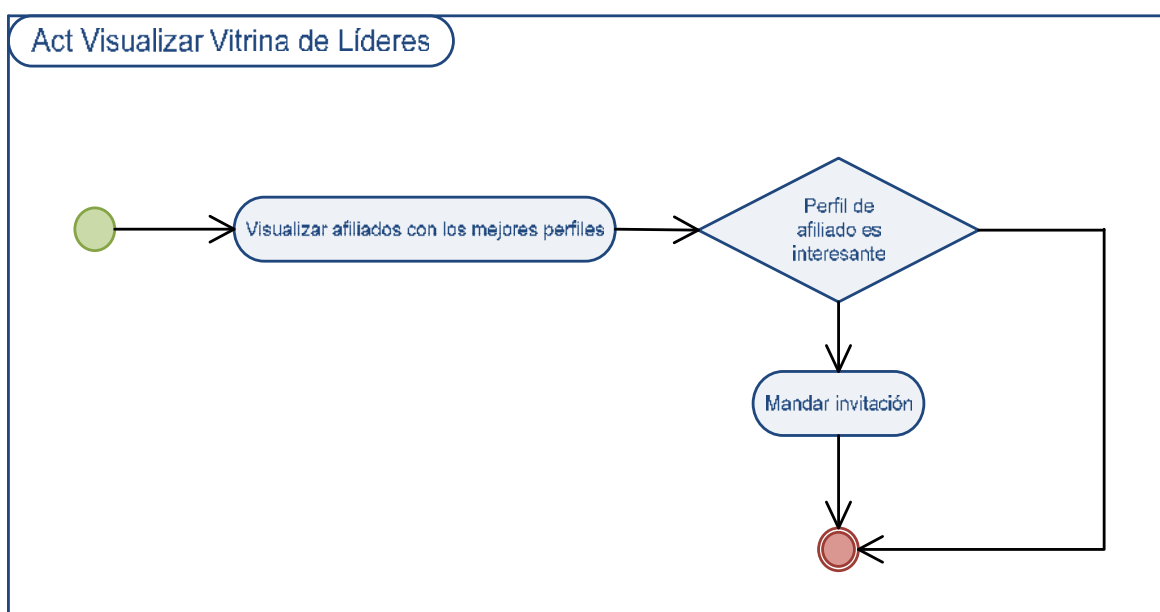


Fig.3-34 Actividades – Vitrina de Líderes

Autores: Tesistas

3.3.2.7. CAMBIAR DE CONTRASEÑA.

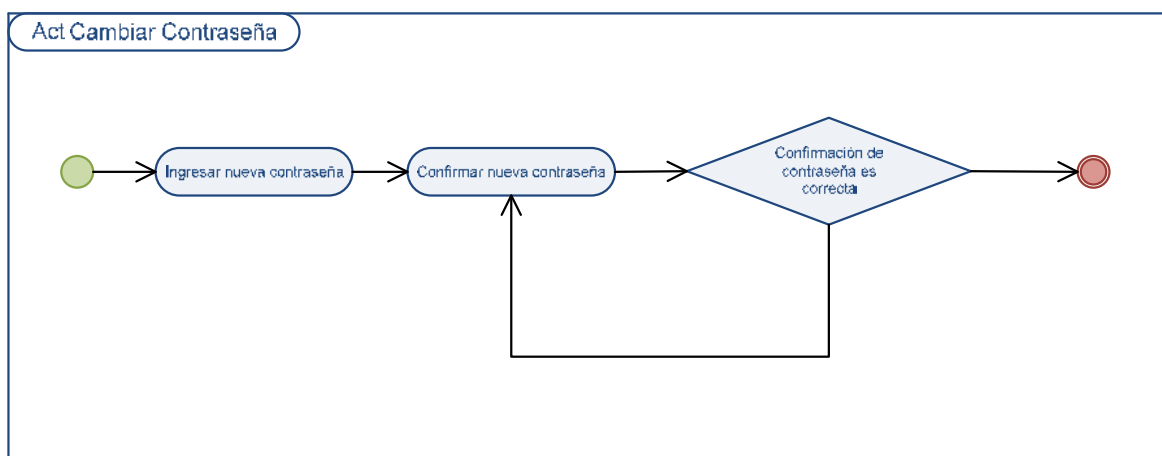


Fig.3.35 Actividades – Cambio de contraseña

Autores: Tesistas

3.3.2.8. VISUALIZAR ESTADÍSTICAS.

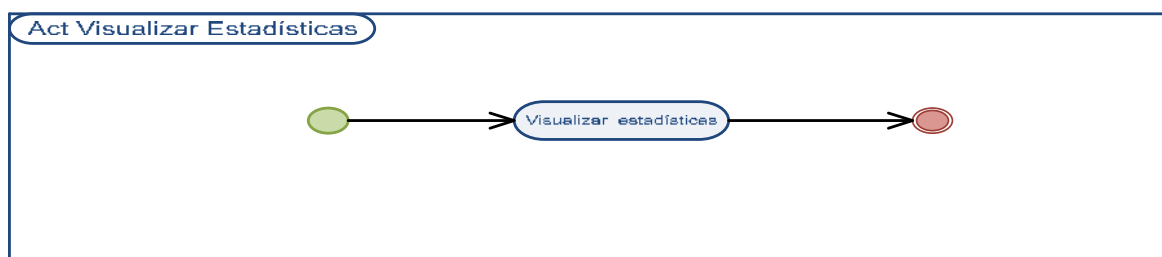


Fig.3.36 Actividades – Visualizar estadísticas.

Autores: Tesistas

3.3.2.9. RESPONDER TICKET.

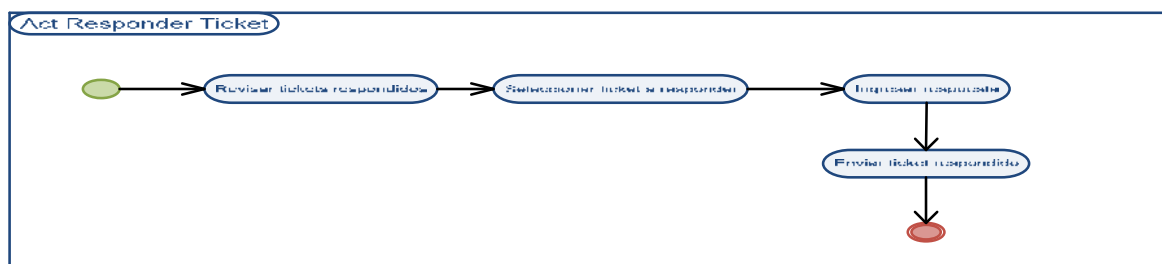


Fig.3.37 Actividades – Responder ticket.

Autores: Tesistas

3.3.2.10. GESTIONAR VECTOR.

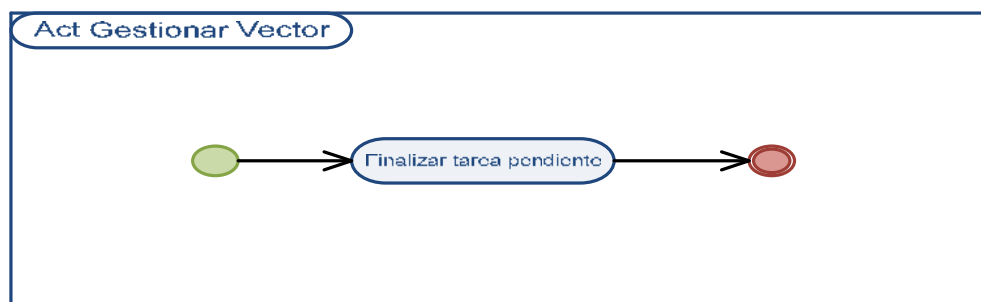


Fig.3.38 Actividades – Gestionar Vector.

Autores: Tesistas

3.3.3. DIAGRAMA NAVEGACIONAL.

3.3.3.1. DIAGRAMA NAVEGACIONAL USUARIO AFILIADO.

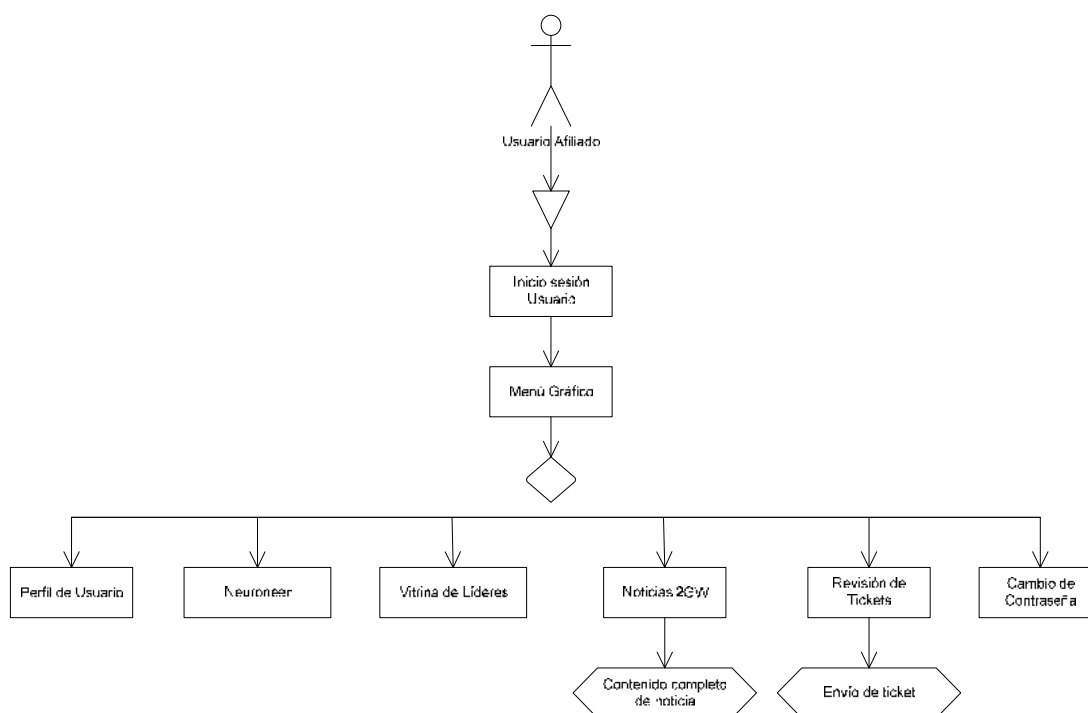


Fig.3.39 Navegacional – Usuario afiliado.

Autores: Tesistas

3.3.3.2. DIAGRAMA NAVEGACIONAL ADMINISTRADOR.

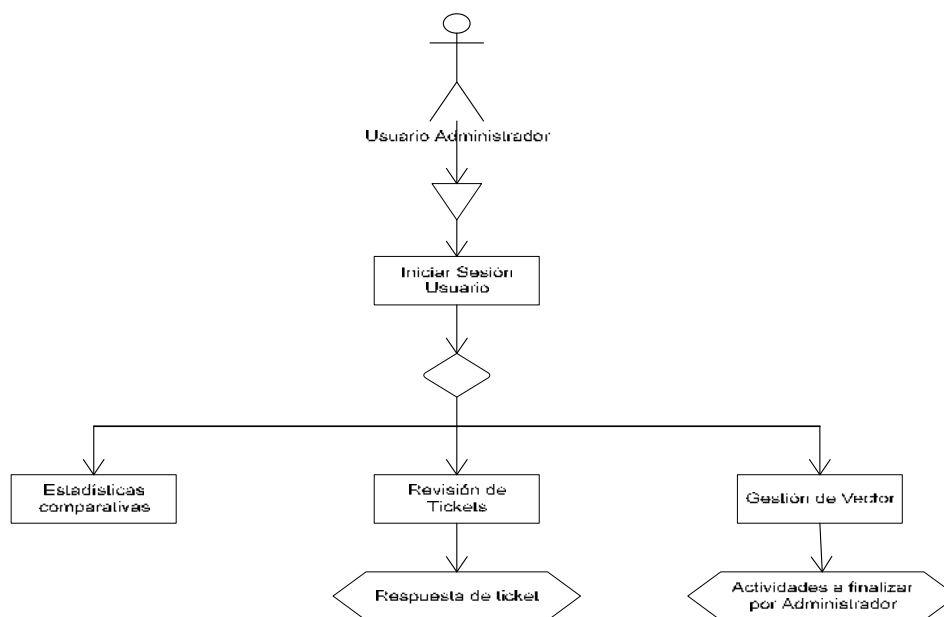


Fig.3.40 Navegacional – Usuario administrador.

Autores: Tesistas

3.3.4. DIAGRAMA DE INTERFAZ.

3.3.4.1. INICIO SESIÓN.

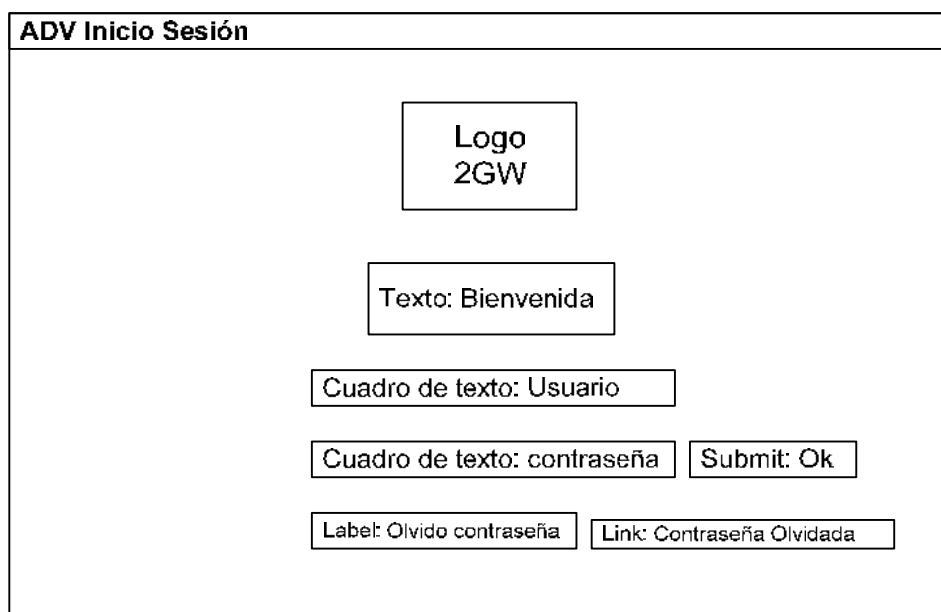


Fig.3.41 Interfaz Abstracta – Inicio Sesión.

Autores: Tesistas

3.3.4.2. MENÚ USUARIO.

ADV Inicio Sesión	
<div>Logo 2GW</div>	
<div>Texto: Bienvenida</div>	
<div>Cuadro de texto: Usuario</div>	
<div>Cuadro de texto: contraseña</div>	<div>Submit: Ok</div>
<div>Label: Olvido contraseña</div>	<div>Link: Contraseña Olvidada</div>

Fig.3.43 Interfaz Abstracta – Inicio Sesión.

Autores: Tesistas

3.3.4.3. NEURONEER.

ADV Neuroneer	
Label: Bienvenido. Nombre usuario	Label: Cerrar Sesión Label: Cambio Idioma
<div>Label: Título Página</div>	
<div> <div>Texto de respuesta de Neuroneer</div> </div>	

Fig.3.44 Interfaz Abstracta – Neuroneer.

Autores: Tesistas

3.3.4.4. VITRINA DE LÍDERES.

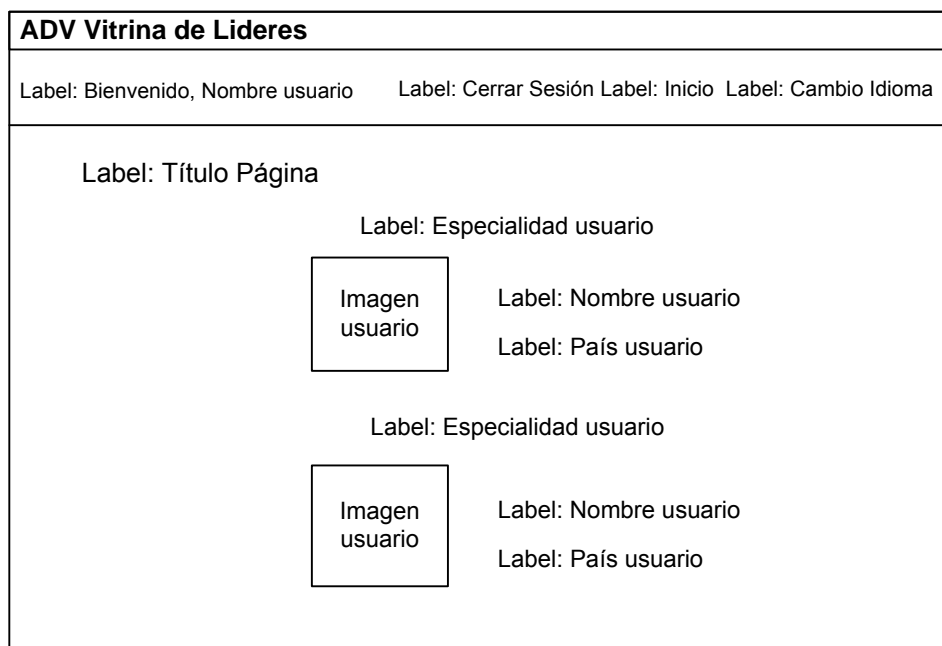


Fig.3.45 Interfaz Abstracta – Vitrina de Líderes.

Autores: Tesistas

3.3.4.5. PERFIL.

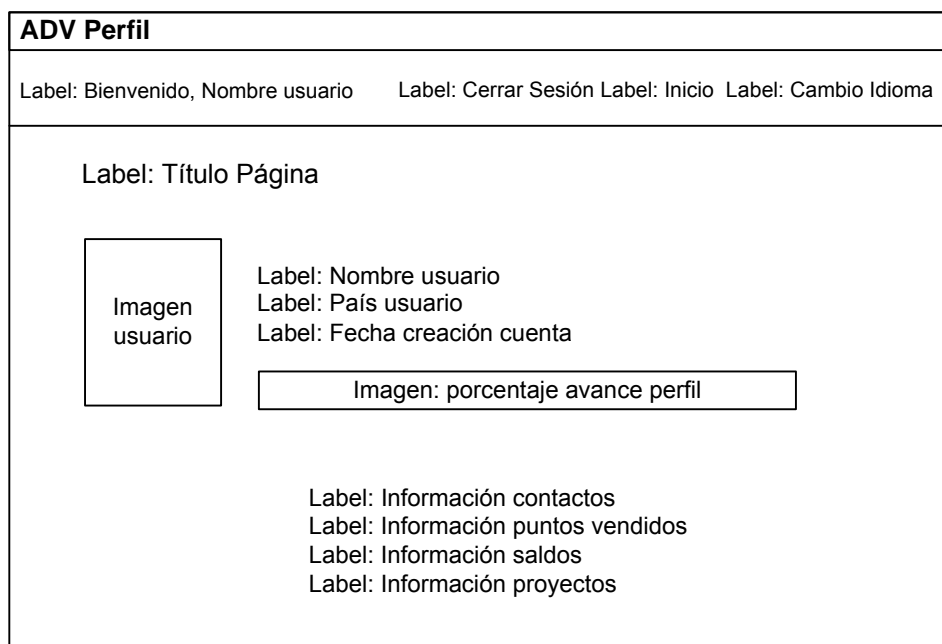


Fig.3.46 Interfaz Abstracta – Perfil.

Autores: Tesistas

3.3.4.6. RESUMEN NOTICIAS.

ADV Resumen Noticias	
Label: Bienvenido, Nombre usuario	Label: Cerrar Sesión Label: Inicio Label: Cambio Idioma
Label: Título Página	
Label: Título noticia Label: Extracto noticia	Label: Fecha noticia <input type="button" value="Link: Tickets"/>
Label: Título noticia Label: Extracto noticia	Label: Fecha noticia <input type="button" value="Link: Tickets"/>
Label: Título noticia Label: Extracto noticia	Label: Fecha noticia <input type="button" value="Link: Tickets"/>

Fig.3.47 Interfaz Abstracta – Resumen noticias.

Autores: Tesistas

3.3.4.7. NOTICIAS.

ADV Noticias	
Label: Bienvenido, Nombre usuario	Label: Cerrar Sesión Label: Inicio Label: Cambio Idioma
Label: Título Noticia	Label: Fecha Noticia
Label: Contenido Noticia	

Fig.3.48 Interfaz Abstracta – Noticias.

Autores: Tesistas

3.3.4.8. CREAR TICKETS.

ADV Crear Tickets			
Label: Bienvenido, Nombre usuario		Label: Cerrar Sesión Label: Inicio Label: Cambio Idioma	
Label: Título Página			
<div>Imagen Link: Nuevo Ticket</div>			
Label: Imagen respuesta	Label: Título ticket	Label: Título Fecha ingreso ticket	Label: Título Fecha respuesta ticket
Imagen respuesta	Link: Título ticket	Label: Fecha ingreso ticket	Label: Fecha respuesta ticket
Imagen respuesta	Link: Título ticket	Label: Fecha ingreso ticket	Label: Fecha respuesta ticket
Imagen respuesta	Link: Título ticket	Label: Fecha ingreso ticket	Label: Fecha respuesta ticket

Fig.3.49 Interfaz Abstracta – Crear tickets.

Autores: Tesistas

3.3.4.9. RESPUESTA DE TICKETS.

ADV Respuesta Tickets	
Label: Bienvenido, Nombre usuario Label: Cerrar Sesión Label: Inicio Label: Cambio Idioma	
Label: Título ticket	Label: Fecha ingreso ticket
<div> <div> <div>Label: Fecha ingreso ticket</div> <div>Área de texto: Ticket enviado</div> </div> <div>▼</div> </div>	
<div> <div> <div>Label: Fecha respuesta ticket</div> <div>Área de texto: Respuesta 2GW</div> </div> <div>▼</div> </div>	

Fig.3.50 Interfaz Abstracta – Revisión de tickets.

Autores: Tesistas

3.3.4.10. NUEVO TICKET.

ADV Nuevo Ticket	
Label: Bienvenido, Nombre usuario	Label: Cerrar Sesión Label: Inicio Label: Cambio Idioma
Label: Título Página	
Label: Título ticket	Cuadro de texto: Título ticket
Label: Categoría ticket	Cuadro de texto: Categoría ticket
Label: Detalle ticket	Área de texto: Detalle ticket
Submit: Enviar ticket	

Fig.3.51 Interfaz Abstracta – Nuevo ticket.

Autores: Tesistas

3.3.4.11. CAMBIO DE CONTRASEÑA.

ADV Cambio de Contraseña	
Label: Bienvenido. Nombre usuario	Label: Cerrar Sesión Label: Cambio Idioma
Label: Título Página	
Cuadro de texto: Contraseña anterior	
Cuadro de texto: Contraseña nueva	
Cuadro de texto: Confirmar contraseña	
Submit: Ok	

Fig.3.52 Interfaz Abstracta – Cambio contraseña.

Autores: Tesistas

3.3.4.12. ESTADÍSTICAS.

ADV Estadísticas		
Label: Bienvenido, Nombre usuario	Label: Cerrar Sesión	Label: Cambio Idioma
<input type="button" value="Submit: Estadísticas"/>	<input type="button" value="Submit: Tickets"/>	<input type="button" value="Submit: Vector"/>
<div>Imagen: Estadísticas</div>		

Fig.3.53 Interfaz Abstracta – Estadísticas.

Autores: Tesistas

3.3.4.13. RESPUESTA DE TICKETS.

ADV Respuesta de Tickets		
Label: Bienvenido, Nombre usuario	Label: Cerrar Sesión	Label: Cambio Idioma
<input type="button" value="Submit: Estadísticas"/>	<input type="button" value="Submit: Tickets"/>	<input type="button" value="Submit: Vector"/>
<div><input type="button" value="Link: Tickets"/></div> <div><input type="button" value="Link: Tickets"/></div> <div><input type="button" value="Link: Tickets"/></div>		

Fig.3.54 Interfaz Abstracta – Respuesta de tickets.

Autores: Tesistas

ADV Respuesta Tickets	
Label: Bienvenido, Nombre usuario Label: Cerrar Sesión Label: Cambio Idioma <div>Submit: Estadísticas</div> <div>Submit: Tickets</div> <div>Submit: Vector</div>	
Label: Nombre usuario:	Cuadro de texto: Nombre usuario
Label: Categoría ticket:	Cuadro de texto: Categoría ticket
Label: Título ticket:	Cuadro de texto: Título ticket
Label: Fecha respuesta:	Cuadro de texto: Fecha ticket
Label: Texto ticket:	Área de texto: Ticket enviado
Label: Respuesta ticket:	Área de texto: Respuesta 2GW
<div>Submit: Responder ticket</div>	

Fig.3.55 Interfaz Abstracta – Respuesta de tickets.

Autores: Tesistas

3.3.4.14. GESTIÓN VECTOR.

ADV Gestión Vector
<div>Submit: Estadísticas</div> <div>Submit: Tickets</div> <div>Submit: Vector</div>
<div>Texto de aviso</div>

Fig.3.56 Interfaz Abstracta – Gestión Vector.

Autores: Tesistas

3.3.7. DIAGRAMA DE DESPLIEGUE.

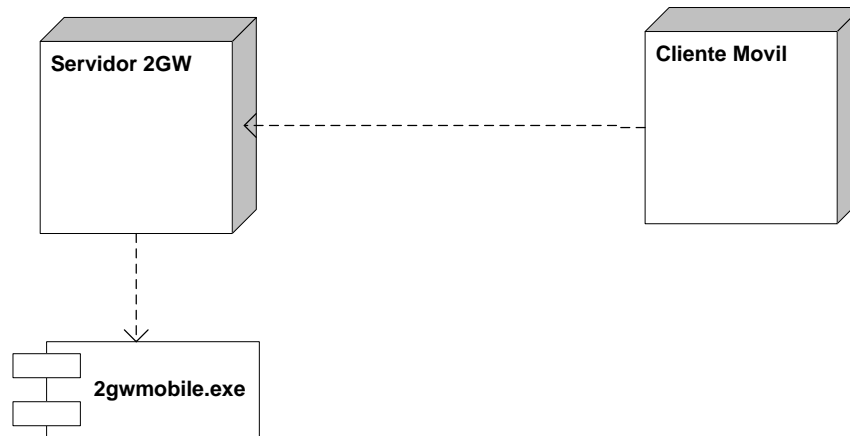


Fig.3.59 Diagrama de Despliegue

Autores: Tesistas

CAPITULO IV

IMPLEMENTACIÓN E IMPLANTACIÓN

4.1 ESTÁNDARES DE PROGRAMACIÓN

4.1.1 COMENTARIOS.

Estándar establecido para todos los comentarios:

'- verifico el estado de sesión

4.1.2 VARIABLES.

Para dar un nombre a una variable siempre se coloca inicialmente la abreviación **var** seguida de las cuatro letras iniciales que hagan referencia al parámetro que se va a manejar, cada segmento de palabra inicia con mayúscula. Ejemplo:

Vamos a hacer referencia al “Código del País” y al “Código de Afiliado”, las variables quedarían de la siguiente forma:

varCodiPais	➔	Código del País
varCodiAfil	➔	Código de Afiliado

4.2 EXPLICACIÓN DE CÓDIGO.

4.2.1 CLASES.

Orden	Nombre Clase	Ubicación	Utilidad	Observación
001	Secure_VeriUsua	00_paginas/ secure_veriusua.asp	Establecer conexión con la BDD, consulta de datos	
002	Index_Wap	02_usuario/ index_wap.asp	Establecer conexión con la BDD	
003	Perfil	02_usuario/ perfil.asp	Establecer conexión con la BDD	
004	Noticias	02_usuario/ service_noti.asp	Establecer conexión con la BDD	
005	Detalle_Noticia	02_usuario/ service_noti_view.as p	Establecer conexión con la BDD	
006	Vitrina	02_usuario/ vitrina.asp	Establecer conexión con la BDD	
007	Detalle_Vitrina	02_usuario/ vitrina.asp	Establecer conexión con la BDD	
008	Tickets	02_usuario/ service_tick.asp	Establecer conexión con la BDD	

Orden	Nombre Clase	Ubicación	Utilidad	Observación
009	Detalle_Tickets	02_usuario/ service_noti_view.asp	Establecer conexión con la BDD	
010	Cambio_Contraseña	02_usuario/ cambio_contr.asp	Establecer conexión con la BDD	

4.2.2 MÉTODOS.

Orden	Nombre Método	Ubicación Método	Utilidad	Observación
001	InicCone()	Secure_VeriUsua Index_Wap Perfil Noticias Detalle_Noticia Vitrina Detalle_Vitrina Tickets Detalle_Tickets Cambio_Contraseña	Iniciar la conexión	
002	TermCone()	Secure_VeriUsua Index_Wap Perfil Noticias Detalle_Noticia Vitrina Detalle_Vitrina Tickets Detalle_Tickets Cambio_Contraseña	Finalizar la conexión	

Orden	Nombre Metodo	Ubicación Metodo	Utilidad	Observación
004	MenuUsua()	Index_Wap	Mostrar el menú para el usuario.	
005	ContNoti()	Noticias	Indica todas las noticias enviadas por 2GW.	
006	DetaNoti()	Noticias	Indica la noticia completa que escogió el usuario.	
007	ConsNoti()	Noticias		
008	ConsMejo()	Vitrina	Muestra información de afiliados más destacados	
009	FiltroMejo()	Vitrina	Muestra información de afiliados más destacados por área de negocio.	
010	ContTick()	Tickets	Cuenta total de tickets del afiliado.	

Orden	Nombre Metodo	Ubicación Metodo	Utilidad	Observación
011	Constick()	Detalle_Tickets	Muestra información del ticket enviado por el afiliado.	
012	CambPass()	Cambio_Contraseña	Cambia la contraseña del usuario.	

Estos artefactos fueron recolectados de acuerdo al RUP en la fase de Elaboración, y son de vital importancia para la siguiente etapa de la notación la cual es la de Construcción.

Cabe recalcar que la construcción de la base de datos fue directamente realizado por la gente de tecnología de 2GETHERWEB, nuestro apoyo con la misma fue en el momento de su diseño, es por esto, que se detallan las siguientes tablas como referencia de un diccionario de datos.

4.3 DICCIONARIO DE DATOS

4.3.1 dat_afiliado.

COLUMNA	TIPO
cod_idioma	smallint(6)
cod_pais	smallint(6)
cod_provinciaestado	smallint(6)
cod_afiliado_padre	int(11)
cod_campaniapublicitaria	smallint(6)

cod_afiliado	int(11)
per_afiliado	char(1)
cla_afiliado	varchar(20)
ema_afiliado	varchar(120)
url_afiliado	varchar(120)
pas_afiliado	varchar(15)
nom_afiliado	varchar(120)
ape_afiliado	varchar(120)
sex_afiliado	char(1)
civ_afiliado	char(1)
fec_naciafiliado	datetime
fec_regiafiliado	datetime
fec_actuaafiliado	datetime
nac_afiliado	varchar(120)
ciu_afiliado	varchar(120)
dom_afiliado	varchar(120)
tel_afiliado	varchar(20)
cel_afiliado	varchar(50)
est_afiliado	char(1)

4.3.2 dat_afiliadocontactoachat

COLUMNA	TIPO
cod_afiliadofuente	int(11)
cod_afiliadodestino	int(11)
cod_chat	int(11)
fec_chat	Datetime
cod_mensaje	smallint(30)
txt_chat_e	varchar(255)
txt_chat_i	varchar(255)
txt_chat_p	varchar(255)

4.3.3 dat_afiliadocontactozona

COLUMNA	TIPO
cod_afiliadofuente	int(11)
cod_afiliadodestino	int(11)
fec_invicontacto	Date
fec_acepcontacto	Date
est_contacto	char(1)

4.3.4 dat_afiliadodatocont

COLUMNA	TIPO
cod_afiliado	int(11)
nom_afiliadodatocont	varchar(120)
ape_afiliadodatocont	varchar(120)
car_afiliadodatocont	varchar(120)
ema_afiliadodatocont	varchar(120)
cel_afiliadodatocont	varchar(20)

4.3.5 dat_afiliadoformprof

COLUMNA	TIPO
cod_especialidad_1	smallint(6)
cod_especialidad_2	smallint(6)
cod_afiliado	int(11)
niv_instafiliadoformprof	char(1)
otr_conoafiliadoformprof	Text

4.3.6 dat_afiliadoingresoconsola

COLUMNA	TIPO
cod_afiliado	int(11)
fec_ingresoconsola	Datetime
num_ingresoconsola	int(11)

4.3.7 dat_afiliadointeres

COLUMNA	TIPO
cod_subambito	smallint(6)
cod_actividad	varchar(10)
cod_compatibilidadaccion	smallint(6)
cod_compatibilidadobjeto	smallint(6)
cod_afiliado	int(11)
cod_interes	smallint(6)
otr_compatibilidadaccion	varchar(30)
otr_compatibilidadobjeto	varchar(30)

4.3.8 dat_afiliadolugatrab

COLUMNA	TIPO
cod_afiliado	int(11)
emp_afiliadolugatrab	varchar(120)
car_afiliadolugatrab	varchar(120)
tie_anioafiliadolugatrab	smallint(6)
tie_meseafiliadolugatrab	smallint(6)

4.3.9 dat_afiliadoneuroneer

COLUMNA	TIPO
cod_afiliado	int(11)
cod_interes_1	smallint(6)
cod_interes_2	smallint(6)

cod_producto	smallint(6)
cod_neuroneer	int(11)
fec_neuroneer	Datetime
ope_neuroneer	varchar(5)
val_neuroneer	Double
tx_neuroneer	varchar(120)
st_neuroneer	varchar(120)
amt_neuroneer	varchar(10)
cc_neuroneer	varchar(10)
cm_neuroneer	varchar(10)
itn_neuroneer	varchar(50)

4.3.1 dat_afiliadoproyecto

COLUMNA	TIPO
cod_subambito	smallint(6)
cod_clasificacionproyecto	smallint(6)
cod_afiliado	int(11)
cod_proyecto	int(11)
fec_regiproyecto	datetime
fec_reviproyecto	datetime
nom_proyecto	varchar(120)
des_proyecto	Text
tip_proyecto	char(1)
otr_proyecto	varchar(120)
mon_proyecto	Double
num_distproyecto	smallint(3)
num_tranproyecto	smallint(3)
mer_proyecto	char(1)
tie_anioproyecto	smallint(3)
tie_meseproyecto	smallint(3)
tie_diasproyecto	smallint(3)

est_proyecto	char(1)
--------------	---------

4.3.2 dat_amigoinvitado

COLUMNA	TIPO
cod_afiliado	int(11)
cod_amigoinvitado	int(11)
fec_enviamigoinvitado	Datetime
ema_amigoinvitado	varchar(120)

4.3.3 dat_comunicacion

COLUMNA	TIPO
cod_afiliado	int(11)
cod_comunicacion	smallint(6)
asu_comunicacion	varchar(120)
tex_comunicacion	Text
opc_expecomunicacion	char(1)
opc_neurcomunicacion	char(1)
opc_benecomunicacion	char(1)
lnk_comunicacion	char(1)
fec_envicomunicacion	Datetime
est_comunicacion	char(1)

4.3.4 dat_noticia

COLUMNA	TIPO
cod_noticia	varchar(11)
fec_noticia	Datetime
tit_noticia_e	varchar(120)
tit_noticia_i	varchar(120)

tit_noticia_p	varchar(120)
txt_cortnoticia_e	Text
txt_cortnoticia_i	Text
txt_cortnoticia_p	Text
txt_largnoticia_e	Text
txt_largnoticia_i	Text
txt_largnoticia_p	Text
est_noticia	char(1)

4.3.5 dat_ticket

COLUMNA	TIPO
cod_afiliado	int(11)
cod_categoriaticket	smallint(6)
cod_ticket	int(11)
fec_regiticket	Datetime
fec_actuticket	Datetime
tit_ticket	varchar(120)
txt_ticket	Text
res_ticket	Text
est_ticket	char(1)

4.3.6 ref_actividad

COLUMNA	TIPO
cod_actividad	varchar(10)
nom_actividad_e	Text
nom_actividad_i	Text
nom_actividad_f	Text
nom_actividad_t	Text
nom_actividad_a	Text
est_actividad	char(2)

des_actividad_e	Text
des_actividad_i	Text
des_actividad_p	Text

4.3.7 ref_cargo

COLUMNA	TIPO
cod_departamento	smallint(6)
cod_cargo	smallint(6)
nom_cargo	char(60)
est_cargo	char(1)

4.3.8 ref_categoriaticket

COLUMNA	TIPO
cod_categoriaticket	smallint(6)
nom_categoriaticket_e	varchar(60)
nom_categoriaticket_i	varchar(60)
nom_categoriaticket_p	varchar(60)

4.3.9 ref_ciudad

COLUMNA	TIPO
cod_ciudad	smallint(6)
ide_ciudad	char(4)
nom_ciudad	char(60)

4.3.10 ref_ciudadreferencia

COLUMNA	TIPO
cod_ciudad	smallint(6)
cod_referencia	smallint(6)

ide_referencia	char(4)
nom_referencia	char(60)

4.3.11 ref_departamento

COLUMNA	TIPO
cod_empresa	smallint(6)
cod_departamento	smallint(6)
ide_departamento	char(4)
nom_departamento	char(60)
des_departamento	char(255)
est_departamento	char(1)

4.3.12 ref_empresa

COLUMNA	TIPO
cod_empresa	smallint(6)
ide_empresa	char(4)
nom_empresa	char(60)
des_empresa	char(255)
est_empresa	char(1)

4.3.13 ref_especialidad

COLUMNA	TIPO
cod_profesion	int(11)
cod_especialidad	int(11)
nom_especialidad_e	varchar(60)
nom_especialidad_i	varchar(60)
nom_especialidad_p	varchar(60)
est_especialidad	char(1)

4.3.14 ref_perfil

COLUMNA	TIPO
cod_sistema	smallint(6)
cod_perfil	smallint(6)
nom_perfil	char(45)
des_perfil	char(255)
opc_panecont_visueven	char(1)
opc_panecont_anuleven	char(1)
opc_panecont_elimeven	char(1)
opc_panecont_retreven	char(1)
opc_geneproc_procwork	char(1)

4.3.15 ref_profesion

COLUMNA	TIPO
cod_profesion	int(11)
nom_profesion_e	varchar(100)
nom_profesion_i	varchar(100)
nom_profesion_p	varchar(100)
tip_profesion	char(1)
est_profesion	char(1)

4.3.16 ref_universidad

COLUMNA	TIPO
cod_idioma	char(2)
cod_pais	char(2)
cod_empresa	decimal(18,0)
cod_kompass	varchar(11)
nom_empresa	varchar(100)
des_empresa	Text

dir_empresa_1	varchar(200)
dir_empresa_2	varchar(200)
pos_empresa	varchar(50)
nom_provinciaestado	varchar(50)
dir_postalempresa_1	varchar(200)
dir_postalempresa_2	varchar(200)
dir_postalempresa_3	varchar(200)
dir_postalempresa_4	varchar(200)
dir_postalempresa_5	varchar(200)
tel_empresa	varchar(15)
fax_empresa	varchar(15)
tlx_empresa	varchar(10)
cif_empresa	varchar(8)
cod_tipoempresa	int(11)
nat_empresa	varchar(20)
fun_empresa	varchar(10)
cap_empresa	varchar(60)
vol_empresa	varchar(60)
pla_empresa	varchar(60)
ban_empresa	varchar(60)
sig_empresa	varchar(60)
sig_sectorempresa	char(3)
nom_contacto	varchar(200)
ema_empresa	varchar(60)
web_empresa	varchar(30)
reg_empresa	varchar(10)
for_juridicaempresa	varchar(50)
est_impoexpoempresa	varchar(10)
fec_actualizacion	Datetime

4.3.17 ref_usuario

COLUMNA	TIPO
amb_usuario	char(10)
cod_cargentiexte	smallint(6)
cod_referencia	smallint(6)
usr_supervisor	char(15)
usr_usuario	char(15)
pas_usuario	char(15)
ced_usuario	char(10)
nom_usuario	char(120)
ape_usuario	char(120)
nom_compusuario	char(120)
ema_usuario	char(120)
sex_usuario	char(1)
est_usuario	char(1)

4.3.18 rel_actividadempresa

COLUMNA	TIPO
cod_actividad	char(10)
cod_empresa	int(11)

4.3.19 rel_afiliadoidioma

COLUMNA	TIPO
cod_afiliado	int(11)
cod_idioma	smallint(6)

4.3.20 rel_afiliadointereszonapais

COLUMNA	TIPO
cod_afiliado	int(11)

cod_interes	smallint(6)
cod_zona	smallint(6)
cod_pais	smallint(6)

4.3.21 rel_carrerauniversidad

COLUMNA	TIPO
cod_carrerauniversidad	int(11)
cod_empresa	decimal(10,0)

4.3.22 rel_compatibilidadaccionobjeto

COLUMNA	TIPO
cod_compatibilidadaccion	smallint(6)
cod_compatibilidadobjeto	smallint(6)

4.3.23 rel_compatibilidadaccionobjetoenlace

COLUMNA	TIPO
cod_compatibilidadaccion_f	smallint(6)
cod_compatibilidadobjeto_f	smallint(6)
cod_compatibilidadaccion_d	smallint(6)
cod_compatibilidadobjeto_d	smallint(6)

4.3.24 rel_comunicacionpais

COLUMNA	TIPO
cod_comunicacion	smallint(6)
cod_pais	smallint(6)

4.3.25 rel_perfilopcionsistema

COLUMNA	TIPO
cod_sistema	smallint(6)
cod_perfil	smallint(6)
cod_opcionsistema	smallint(6)

4.3.26 rel_usuarioperfil

COLUMNA	TIPO
usr_usuario	char(15)
cod_sistema	smallint(6)
cod_perfil	smallint(6)

4.3.27 sis_afiliacionvali

COLUMNA	TIPO
cod_afiliacionvali	int(11)
fec_afiliacionvali	datetime
txt_afiliacionvali	varchar(6)

4.3.28 sis_ambito

COLUMNA	TIPO
cod_ambito	char(1)
nom_ambito_e	char(30)
nom_ambito_i	char(30)
nom_ambito_p	char(30)

4.3.29 sis_ambitoproyecto

COLUMNA	TIPO
cod_ambitoproyecto	smallint(6)
nom_ambitoproyecto_e	varchar(60)

nom_ambitoproyecto_i	varchar(60)
nom_ambitoproyecto_p	varchar(60)
ord_ambitoproyecto	smallint(6)

4.3.30 sis_clasificacionproyecto

COLUMNA	TIPO
cod_clasificacionproyecto	smallint(6)
nom_clasificacionproyecto_e	varchar(60)
nom_clasificacionproyecto_i	varchar(60)
nom_clasificacionproyecto_p	varchar(60)

4.3.31 sis_compatibilidadaccion

COLUMNA	TIPO
cod_compatibilidadaccion	smallint(6)
nom_compatibilidadaccion_e	varchar(60)
nom_compatibilidadaccion_i	varchar(60)
nom_compatibilidadaccion_p	varchar(60)

4.3.32 sis_compatibilidadobjeto

COLUMNA	TIPO
cod_compatibilidadobjeto	smallint(6)
nom_compatibilidadobjeto_e	varchar(60)
nom_compatibilidadobjeto_i	varchar(60)
nom_compatibilidadobjeto_p	varchar(60)

4.3.33 sis_continente

COLUMNA	TIPO
---------	------

cod_continente	smallint(6)
nom_continente_e	varchar(120)
nom_continente_i	varchar(120)
nom_continente_p	varchar(120)

4.3.34 sis_continentezona

COLUMNA	TIPO
cod_continente	smallint(6)
cod_zona	smallint(6)
nom_zona_e	varchar(120)
nom_zona_i	varchar(120)
nom_zona_P	varchar(120)
fus_internaID	char(2)
fus_flashname	varchar(60)

4.3.35 sis_empresa

COLUMNA	TIPO
cod_idioma	char(2)
cod_pais	char(3)
cod_empresa	decimal(10,0)
cod_kompass	varchar(11)
nom_empresa	varchar(100)
des_empresa	text
dir_empresa_1	varchar(200)
dir_empresa_2	varchar(200)
pos_empresa	varchar(50)
nom_provinciaestado	varchar(50)
dir_postalempresa_1	varchar(200)
dir_postalempresa_2	varchar(200)
dir_postalempresa_3	varchar(200)

dir_postalempresa_4	varchar(200)
dir_postalempresa_5	varchar(200)
tel_empresa	varchar(15)
fax_empresa	varchar(15)
tlx_empresa	varchar(10)
cif_empresa	varchar(8)
cod_tipoempresa	int(11)
nat_empresa	varchar(20)
fun_empresa	varchar(10)
cap_empresa	varchar(60)
vol_empresa	varchar(60)
pla_empresa	varchar(60)
ban_empresa	varchar(60)
sig_empresa	varchar(60)
sig_sectorempresa	char(3)
cod_sectorempresa	varchar(60)
nom_contacto	varchar(200)
ema_empresa	varchar(60)
web_empresa	varchar(30)
reg_empresa	varchar(10)
for_juridicaempresa	varchar(50)
est_impoexpoempresa	varchar(10)
fec_actualizacion	datetime

4.3.36 sis_encuesta

COLUMNA	TIPO
cod_encuesta	tinyint(3)
nom_encuesta	varchar(120)

4.3.37 sis_encuestapregunta

COLUMNA	TIPO
cod_encuesta	smallint(3)
cod_pregunta	smallint(3)
tip_pregunta	char(1)
pre_pregunta	varchar(255)
res_pregunta_1	varchar(120)
res_pregunta_2	varchar(120)
res_pregunta_3	varchar(120)

4.3.38 sis_encuestapreguntaafiliado

COLUMNA	TIPO
cod_encuesta	smallint(3)
cod_pregunta	smallint(3)
cod_afiliado	int(3)
val_respuesta_1	varchar(255)
val_respuesta_2	varchar(255)

4.3.39 sis_idioma

COLUMNA	TIPO
cod_idioma	char(2)
nom_idioma_e	char(30)
nom_idioma_i	char(30)
nom_idioma_p	char(30)

4.3.40 sis_landingpageacceso

COLUMNA	TIPO
cod_landingpage	smallint(6)
cod_idioma	smallint(6)
fec_acceso	datetime

hor_acceso	smallint(6)
num_acceso	int(11)

4.3.41 sis_matrizvalidacion

COLUMNA	TIPO
cod_matrizvalidacion	smallint(6)
car_matrizvalidacion	char(1)
nom_matrizvalidacion	varchar(10)

4.3.42 sis_mensaje

COLUMNA	TIPO
cod_mensaje	smallint(3)
nom_mensaje_e	varchar(120)
nom_mensaje_i	varchar(120)
nom_mensaje_p	varchar(120)

4.3.43 sis_numeroempresasbase

COLUMNA	TIPO
cod_actividad	char(10)
cod_pais	smallint(6)
num_empresas	smallint(6)

4.3.44 sis_pais

COLUMNA	TIPO
cod_zona	smallint(6)
cod_pais	smallint(6)
ide_pais	char(2)
nom_pais_e	varchar(60)

nom_pais_i	varchar(60)
nom_pais_p	varchar(60)
nac_pais_e	varchar(60)
nac_pais_i	varchar(60)
nac_pais_p	varchar(60)
cod_areapais	char(3)
fus_internallID	varchar(5)
fus_shortname	varchar(5)
fus_flashname	varchar(60)

4.3.45 sis_productoneuroneer

COLUMNA	TIPO
cod_producto	smallint(6)
ide_producto	varchar(10)
nom_producto	varchar(60)
des_producto	tinytext
pre_producto	double

4.3.46 sis_provinciaestado

COLUMNA	TIPO
cod_pais	smallint(6)
cod_provinciaestado	smallint(6)
nom_provinciaestado	varchar(120)
cod_areaprovinciaestado	varchar(6)

4.3.47 sis_sistema

COLUMNA	TIPO
cod_sistema	smallint(6)
nom_sistema	varchar(60)

4.3.48 sis_sistemamodulo

COLUMNA	TIPO
cod_sistema	smallint(6)
cod_modulosistema	smallint(6)
nom_modulosistema	varchar(60)
ord_modulosistema	smallint(6)

4.3.49 sis_sistemaopcion

COLUMNA	TIPO
cod_sistema	smallint(6)
cod_modulosistema	smallint(6)
cod_opcionsistema	smallint(6)
nom_opcionsistema	char(45)
url_opcionsistema	char(120)
tip_opcionsistema	char(1)
ord_opcionsistema	smallint(6)

4.3.50 sis_subambito

COLUMNA	TIPO
cod_ambito	char(1)
cod_subambito	smallint(6)
nom_subambito_e	varchar(120)
nom_subambito_i	varchar(120)
nom_subambito_p	varchar(120)

4.3.51 sis_subambitoactividad

COLUMNA	TIPO
----------------	-------------

cod_subambito	smallint(6)
cod_actividad	char(10)

4.3.52 sis_terminopagina

COLUMNA	TIPO
cod_terminopagina	int(11)
val_terminopagina_e	Text
val_terminopagina_i	Text
val_terminopagina_p	Text

4.4 REGISTRO DE CONSULTAS (PRUEBA DE LA BASE DE DATOS).

Código	Sentencia	Utilidad	Inf. devuelta	Observaciones
001	SELECT * FROM dat_afiliado WHERE ema_afiliado="" & varUserUsua & ""	Obtener los datos del afiliado que inicio sesión	Registros	Ok
002	SELECT num_ingresoconsola FROM dat_afiliadoingresoconsola WHERE cod_afiliado="" & Session("Codi_Usua") & " AND fec_ingresoconsola="" & varFechActu & ""	Obtener información del usuario para guardar estadísticas de ingreso	Registros	Ok
003	UPDATE dat_afiliadoingresoconsola SET num_ingresoconsola=num_ingreso consola+1 WHERE cod_afiliado="" & Session("Codi_Usua") & " AND	Guardar estadísticas de ingreso	Información almacenada	Ok

	fec_ingresoconsola="" & varFechActu & ""			
Código	Sentencia	Utilidad	Inf. devuelta	Observaciones
004	INSERT INTO dat_afiliadoingresoconsola VALUES (" & Session("Codi_Usua") & "," & varFechActu & ",1)")	Guardar datos del usuario	Información almacenada	Ok
005	SELECT A.*, B.nom_pais_ & varldioLetrUsua & " AS nom_pais FROM dat_afiliado A, sis_pais B WHERE A.ema_afiliado="" & Session("User_Usua") & "" AND A.cod_pais=B.cod_pais	Consultar los datos del afiliado para cambiar contraseña	Password de usuario	Ok
006	"UPDATE dat_afiliado SET nom_afiliado="" & varContNuev & "" WHERE ema_afiliado='verokasol85@hotmail.com'	Actualizar la nueva contraseña ingresada		Ok
007	SELECT * FROM dat_noticia WHERE est_noticia='A' ORDER BY fec_noticia DESC, cod_noticia DESC	Selecciona las noticias activas	Datos	Ok
008	SELECT * FROM dat_noticia WHERE cod_noticia="" & varCodiNoti	Visualizar la noticia complete	Datos	Ok
009	SELECT MAX(cod_ticket) AS cod_ticket FROM dat_ticket	Obtener el ultimo código de ticket	Datos	Ok
010	INSERT INTO dat_ticket VALUES	Ingresar el		Ok

	(" & Session("Codi_Usua") & "," & varCodiCate & "," & varCodiTick & "," & varFechRegiTick & ",null," & varTituTick & "," & varTextTick & ",null,'A')	nuevo ticket		
Código	Sentencia	Utilidad	Inf. devuelta	Observaciones
011	SELECT * FROM dat_ticket WHERE cod_afiliado=" & Session("Codi_Usua") & " ORDER BY fec_regiticket DESC, cod_ticket DESC	Visualizar todos los tickets enviados por el usuario	Datos	Ok
012	SELECT A.*, A.sex_afiliado, B.nom_pais_" & varldioLetrUsua & ", D.nom_especialidad_" & varldioLetrUsua & " FROM dat_afiliado A, sis_pais B, dat_afiliadoformprof C, ref_especialidad D WHERE A.cod_pais=B.cod_pais AND A.cod_afiliado=C.cod_afiliado AND C.cod_especialidad_1=D.cod_especialidad AND A.est_afiliado='A' AND A.cod_afiliado IN (SELECT cod_afiliado FROM dat_afiliadoneuroneer) ORDER BY A.nom_afiliado, A.ape_afiliado LIMIT 2	Recuperar todos los contactos del usuario	Datos	Ok
013	SELECT B.*, C.nom_subambito_e, E.cod_compatibilidadaccion, E.nom_compatibilidadaccion_e, F.cod_compatibilidadobjeto, F.nom_compatibilidadobjeto_e,	Selecciona los expedientes del afiliado	Datos	Ok

	G.nom_actividad_e FROM dat_afiliadointeres B, sis_subambito C, sis_compatibilidadaccion E, sis_compatibilidadobjeto F, ref_actividad G WHERE B.cod_afiliado=" & recordset("cod_afiliado") & " AND B.cod_actividad=G.cod_actividad AND B.cod_compatibilidadobjeto=F.cod _compatibilidadobjeto AND B.cod_compatibilidadaccion=E.cod _compatibilidadaccion AND B.cod_subambito=C.cod_subambit o			
014	SELECT cod_afiliadodestino FROM dat_afiliadocontactozona WHERE cod_afiliadofuente=" & Session("Codi_Usua") & " AND cod_afiliadodestino=" & recordset("cod_afiliado")	Verifica si el afiliado tiene solicitudes	Datos	Ok
015	SELECT D.nom_especialidad_" & varldioLetrUsua & " FROM dat_afiliadoformprof C, ref_especialidad D WHERE C.cod_especialidad_1=D.cod_esp ecialidad AND C.cod_afiliado=" & recordset("cod_afiliado")	Obtiene la profesión del afiliado	Datos	Ok
016	SELECT A.*, B.cod_ambito FROM dat_afiliadointeres A, sis_subambito B WHERE A.cod_afiliado=" &	Selecciona los intereses ingresados por el afiliado	Datos	Ok

	Session("Codi_Usua") & " AND A.cod_subambito=B.cod_subambito			
Código	Sentencia	Utilidad	Inf. devuelta	Observaciones
017	SELECT cod_zona, cod_pais FROM rel_afiliadointereszonapais WHERE cod_afiliado=" & Session("Codi_Usua") & " AND cod_interes=" & varCodiInte	Recupera la lista de países y zonas	Datos	Ok
018	SELECT COUNT(DISTINCT A.cod_afiliado) AS cuenta FROM dat_afiliado A, sis_pais B, dat_afiliadoformprof C, ref_especialidad D, dat_afiliadointeres E, rel_afiliadointereszonapais F WHERE A.cod_afiliado<>" & Session("Codi_Usua") & " AND A.cod_pais=B.cod_pais AND A.cod_afiliado=C.cod_afiliado AND C.cod_especialidad_1=D.cod_especialidad AND A.cod_afiliado=E.cod_afiliado AND E.cod_afiliado=F.cod_afiliado AND E.cod_interes=F.cod_interes AND E.cod_compatibilidadaccion=" & varCodiCompAcci & " AND E.cod_compatibilidadobjeto=" & varCodiCompObje & " AND A.est_afiliado='A' " & varFiltAuxiSuba & varFiltAuxiPais & " ORDER BY A.nom_afiliado,	Cuenta el numero de contactos encontrados	Datos	Ok

	A.ape_afiliado			
019	SELECT nom_proyecto FROM dat_afiliadoproyecto WHERE cod_afiliado=" & recordset("cod_afiliado")	Verifica si el afiliado tiene algún proyecto	Datos	Ok
020	SELECT * FROM dat_afiliado WHERE cod_afiliado=" & Session("Codi_Usua")	Recupera los datos del afiliado que envo invitacion	Datos	Ok
021	INSERT INTO dat_afiliadocontactozona VALUES (" & Session("Codi_Usua") & "," & recordset("cod_afiliado") & "," & varFechActu & "',null,'P'	Ingresa el registro de invitación en estado pendiente	Datos	Ok
022	SELECT * FROM sis_pais WHERE cod_pais IN (SELECT DISTINCT cod_pais FROM sis_provinciaestado) ORDER BY nom_pais_" & varldioLetrUsua	Selecciona informacion de los paises para realizar grafica estadistica	Datos	Ok
023	SELECT A.cod_pais, A.nom_pais_" & varldioLetrUsua & " AS nom_pais, (COUNT(B.cod_empresa) + (SELECT SUM(C.num_empresas) FROM sis_numeroempresasbase C WHERE C.cod_pais=A.cod_pais)) AS cuenta FROM sis_pais A, sis_empresa B WHERE A.cod_pais=B.cod_pais GROUP	Realiza la grafia estadistica	Datos	Ok

	BY A.cod_pais, A.nom_pais_" & varIdioLetrUsua & " ORDER BY cuenta DESC LIMIT 10			
--	---	--	--	--

4.5 PRUEBAS

En esta etapa del proyecto se evidencia la importancia de la fase de Construcción y Transición del RUP. Ya que el desarrollo del sistema mediante las pruebas garantiza que el software esté listo para los usuarios iniciales.

4.5.1 PRUEBA DE FUNCIONALIDAD

4.5.1.1 READY.MOBI

Es una herramienta que nos permite comprobar, mediante una gran cantidad tests específicos, si nuestro sitio web es accesible correctamente desde un teléfono móvil.

La dirección de esta pagina es; http://ready.mobi/launch.jsp?locale=en_EN, esta página utiliza una puntuación del 1 al 5, siendo 1 y 2 una puntuación mala, 3 tiene como calificación justa y los números 4 y 5 .

A continuación daremos una vista completa de las pruebas que se realizaron con nuestra aplicación WAP.

4.5.1.1.1 Login de acceso



Fig. 4.1 Login de acceso a mobiReady

Autor: Tesistas

Fuente: mobiReady

Antes de ingresar a las funcionalidades de esta herramienta y para poder aprovechar sus servicios es necesario registrarse, este es un proceso muy sencillo lo único que requieren es: e-mail, un username para identificarse, y listo, después de este proceso se envía un correo electrónico el cual llega hasta la bandeja de entrada del e-mail antes especificado; en este correo tendremos la bienvenida a la herramienta y el user con su respectiva contraseña.

4.5.1.1.2 *Página Principal*

Esta página principal nos indica que tipos de análisis se pueden realizar y los divide por categorías:

- Page Test
- Markup Test
- Site Test

Para los intereses de nuestra aplicación se tomaron como referencia la “Page Test” y “Site Test”.

Sólo tenemos que ingresar la URL del sitio web que queremos comprobar y el sistema se encarga de realizar los tests necesarios, luego nos da la puntuación y un completo informe con datos para analizar, ya sean puntos a favor y errores acompañados de recomendaciones.

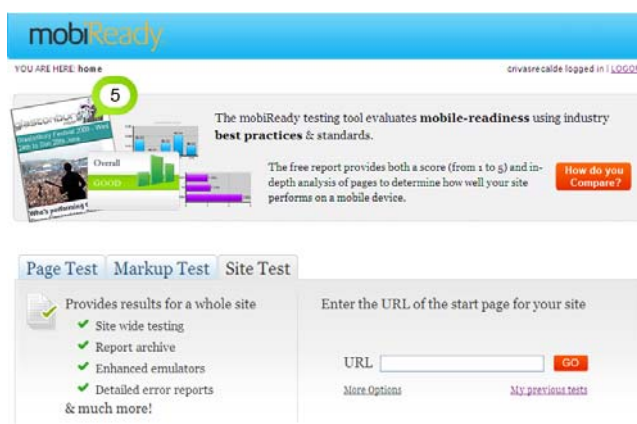


Fig. 4.2 Página Principal

Autor: Tesistas

Fuente: mobiReady

4.5.1.1.3 Page Test

La calificación de nuestro portal es de 4 puntos una calificación buena; en este test además podemos ver estadísticas como el tamaño de la página, el costo estimado y la velocidad estimada.

El tamaño es de 21.32k que es una calificación justa, tomando en cuenta que nuestro portal muestra foto del cliente en el instante que se loguea e ingresa a la sección de perfil. Obviamente esta es la causa por la cual el tamaño es un poco elevado.

Page results

URL tested: mobile.2gwvirtual.com

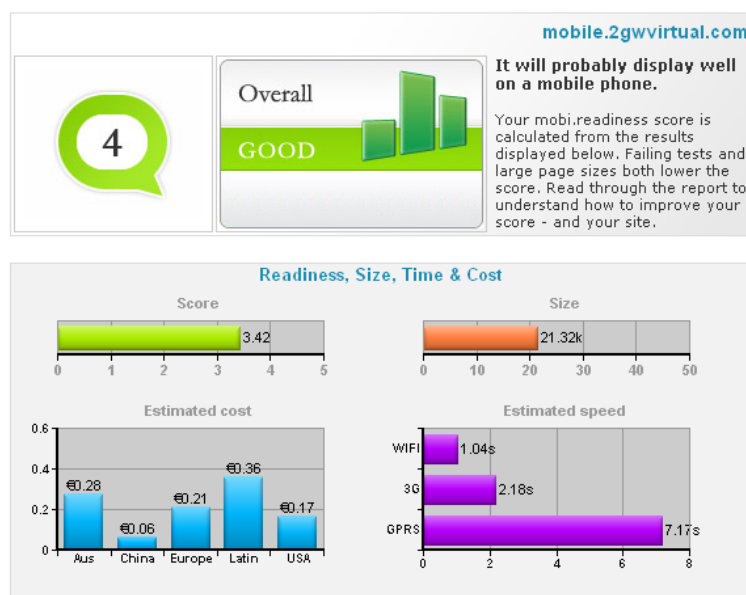


Fig. 4.3 Visualización del Page Test

Autor: Tesistas

Fuente: mobiReady

El costo estimado del portal WAP se visualiza en el gráfico estadístico que se encuentra en el lado inferior izquierdo de la figura 4.3, podemos ver que en diferentes regiones del mundo el portal tiene un coste específico cotizado en euros.

En la parte inferior derecha de la figura 4.3, podemos ver la velocidad estimada del portal, haciendo comparaciones de la tecnología contra el tiempo de ejecución del portal. Así, para un teléfono móvil con tecnología Inalámbrica WIFI la velocidad en tiempo de ejecución es de 1.04s, para un teléfono con tecnología 3G el tiempo es de 2.18s y finalmente para un celular con tecnología GPRS el tiempo que la herramienta definió es de 7.17s.

4.5.1.1.4 Site Test

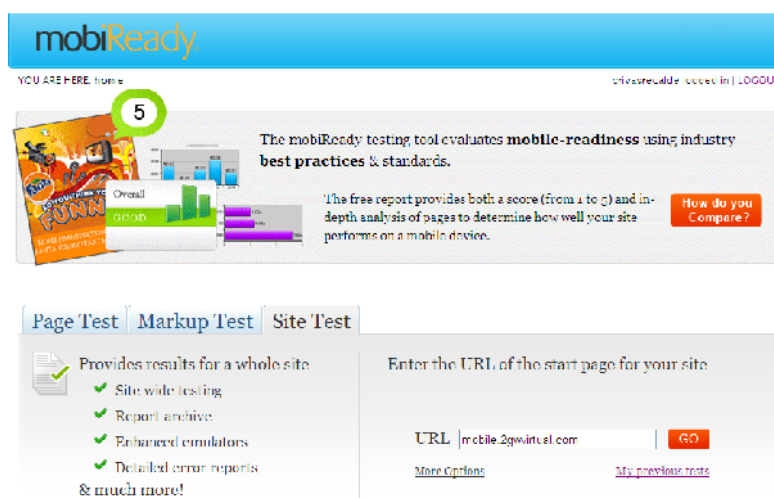


Fig. 4.4 Visualización del Site Test

Autor: Tesistas

Fuente: mobiReady

Al igual que en el análisis “Page Test” es necesario ingresar el URL de la pagina que vamos a analizar en nuestro caso son 2 dominios estos son:

- Mobile.2gwvirtual.com
- Mobileadmin.2gwvirtual.com

4.5.1.1.5 Resultados De Los Test Realizados

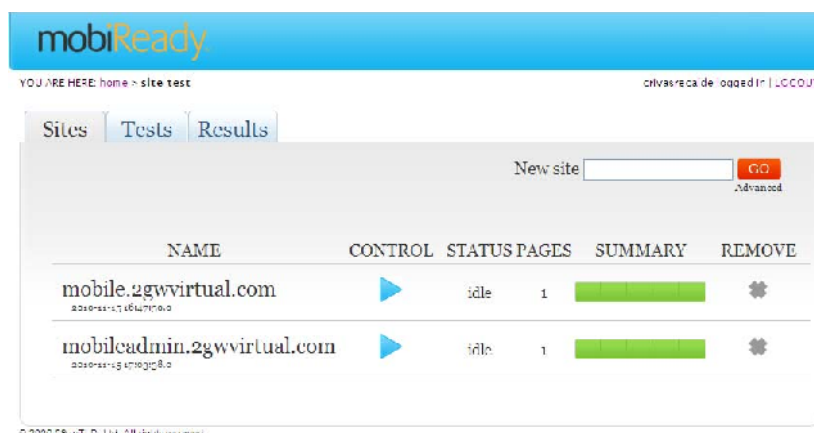


Fig. 4.5 Visualización de página Site

Autor: Tesistas

Fuente: mobiReady

En la figura 4.5 podemos visualizar el nombre del o los sitios analizados, el control del registro; es decir, si el programa fue ejecutado se muestra un símbolo de play



, en caso de que el o los sitios no hayan sido corridos se muestra el siguiente

símbolo .

4.5.1.1.6 Resultados De Los Sitios

URL	SCORE	SIZE(K)	VALID XHTML	2nd LEVEL DOMAIN	NO FRAMES	START TIME
http://mobileadmin.2gwvirtual.com	4	21.32	✓	✓	✓	17:03:38.0
http://mobile.2gwvirtual.com	4	21.32	✓	✓	✓	16:47:50.0

Showing 0-2 of 2 page results
[first](#) - [prev](#) - [next](#) - [last](#) -

Fig. 4.6 Visualización de Resultados

Autor: Tesistas

Fuente: mobiReady

4.5.1.1.7 Resultado Individual: Mobileadmin

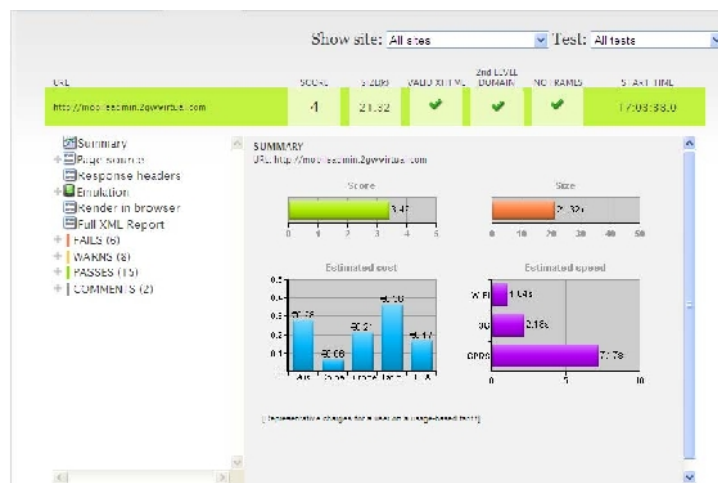


Fig. 4.7 Detalle Resultado Individual Mobileadmin

Autor: Tesistas

Fuente: mobiReady

En la figura 4.7 podemos constatar q las pruebas han sido exitosas, se detallan los puntos que se analizaron en este test y una vez más las pruebas han sido pasadas satisfactoriamente.

4.5.1.1.8 Resultado Individual: Mobile



Fig. 4.8 Detalle Resultado Individual mobile

Autor: Tesistas

Fuente: mobiReady

4.5.2 PRUEBAS DE CARGA

Las pruebas de carga se realizan para tener una idea del rendimiento del servidor con respecto a la carga de datos que tiene el portal WAP, A continuación detallamos algunos de los reportes gráficos que realizamos para poder constatar el correcto funcionamiento de transferencia de datos, ahorro de clics y ancho de banda del servidor.

4.5.2.1 Transferred Data & System & CPU Load

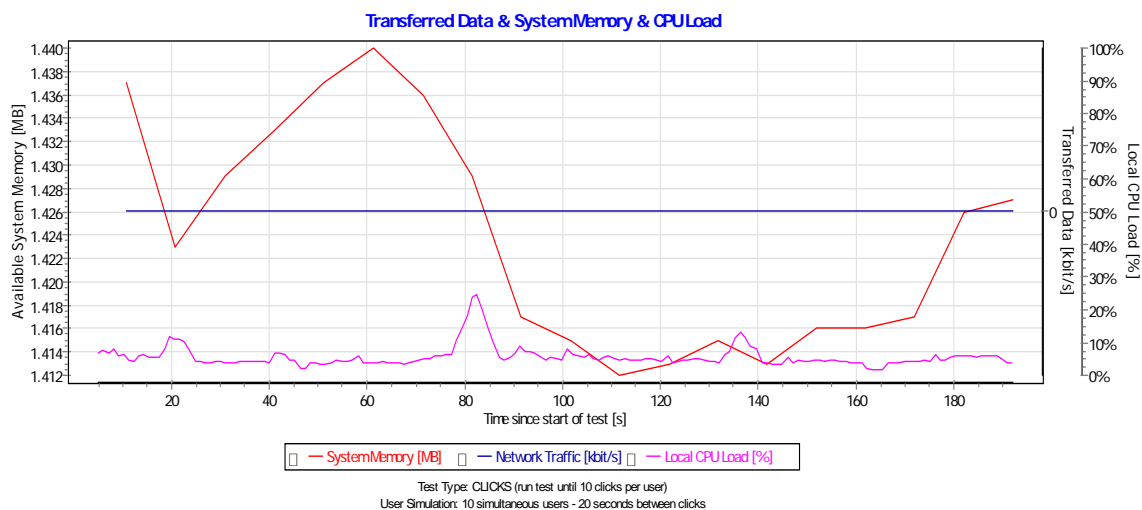


Fig. 4.9 Reporte de carga 1
Fuente: Webserver Stress Tool

Notamos que la mayor parte del trabajo lo realiza la memoria del servidor, dejando un porcentaje mínimo de uso del CPU.

4.5.2.2 Reporte RAMP – Open Requests and Transferred Data

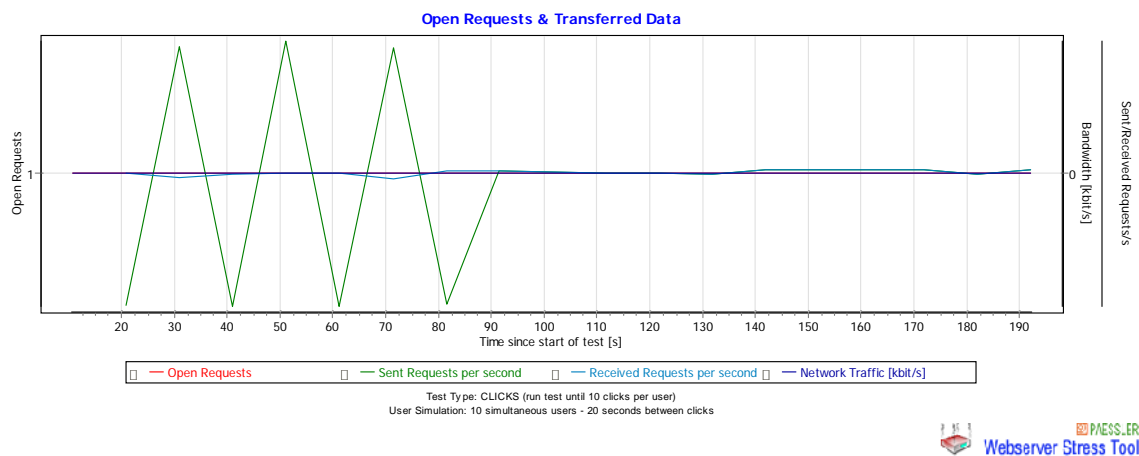


Fig. 4.10 Reporte de carga 2

Fuente: Webserver Stress Tool

En el instante que se realizan las peticiones en un tiempo determinado y tomando en cuenta el número de usuarios que ingresan al portal se concluye que el tráfico de carga en la red es constante y continuo en el tiempo.

4.5.2.3 Reporte TIME – Click Time, Hits/s, Users/s (all URLs)

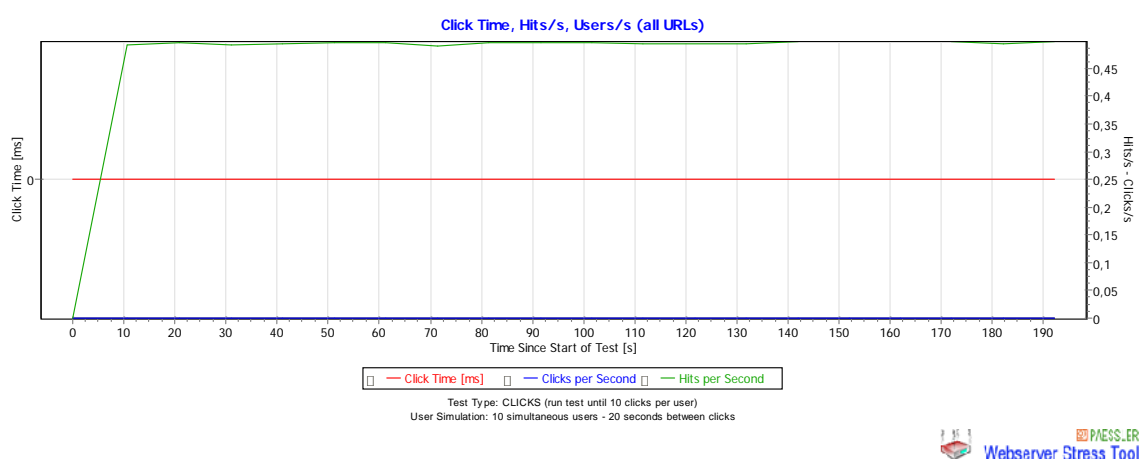


Fig. 4.11 Reporte de carga 3

Fuente: Webserver Stress Tool

Como promedio de clics por usuario versus el tiempo de ejecución obtenemos un rango de 0,40s, el cual es un tiempo de respuesta considerable para la aplicación WAP.

4.5.2.4 Server and User Bandwith

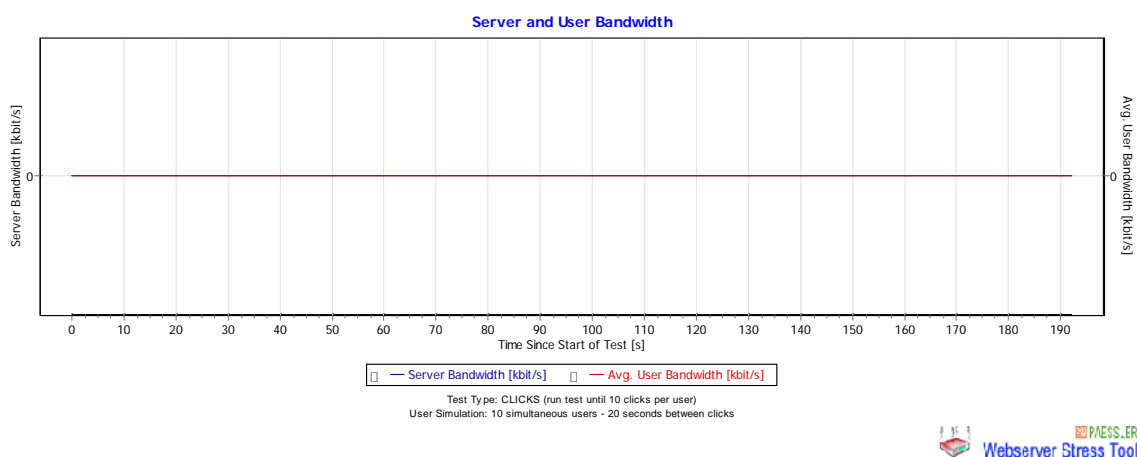


Fig. 4.12 Reporte de carga 4

Fuente: Webserver Stress Tool

El promedio de carga e impacto que tiene el portal WAP con respecto al ancho de banda que utiliza en el servidor dedicado es menor a 1s, es decir, que la ejecución del sistema es confiable en cuestiones de rendimiento del servidor

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La aplicación móvil fue desarrollada en base a las funcionalidades y requerimientos inicialmente acordadas, sin embargo, en el proceso de análisis y diseño, algunos de los requerimientos iniciales fueron eliminados, ya que la organización considero que no debían ser implementados en el portal por razones propias de la misma.
- Los recursos de 2GETHERWEB en ningún momento se vieron afectados por la realización de este proyecto, lo que quiere decir que no se incurrieron en gastos financiados por la empresa.
- Los manuales necesarios para el correcto manejo del sistema fueron redactados de una manera simple y de fácil entendimiento para el usuario afiliado y el administrador, esto no significa, que se hayan omitido pasos importantes, ya que, la explicación de su funcionamiento se define en pasos sencillos.
- El primer paso que se tiene que dar para mejorar la eficiencia operacional de una organización, consiste en definir claramente los procesos. En el caso del diseño de software es también importante. El Modelador de Procesos BPMN Bizagi, permite diagramar y documentar los procesos de la manera más rápida y fácil posible.
- UML fue de vital importancia en el capítulo de diseño del portal, los diagramas ayudaron a entender de mejor manera el funcionamiento del sistema.
- RUP es bastante útil, ya que, cada fase nos ayuda a analizar y documentar de forma clara el sistema, permitiendo mejorar la calidad de los entregables y por ende el proyecto en su conjunto.

- Los entregables de cada una de las fases del RUP son de suma importancia, porque nos permiten tener un seguimiento real del avance del proyecto.
- La programación WAP, debe ser sencilla, muy ligera en relación al consumo de recursos, debido a que la finalidad de la misma es brindar una interfaz fácil de usar, no muy recargada visualmente; centrándose en la funcionalidad principal que la de mantener informado de las actividades procesadas al usuario afiliado y administrador.
- Las pruebas realizadas al portal están acorde a las necesidades de funcionalidad, carga y usabilidad requeridas por el patrocinador (sponsor) del proyecto certificando que el portal se encuentra operable en su totalidad.
- BPMN es una notación bastante integral que nos facilita el entendimiento de los procesos de negocio que se realizan en 2GETHERWEB.
- El modelador de procesos BPMN(BIZAGI) es definitivamente la mejor herramienta para definir el desarrollo de las actividades de las organizaciones empresariales, esto desde el enfoque utilizado en nuestro proyecto.
- Se utilizaron los diagramas UML que tenían mayor relevancia en el desarrollo del proyecto.
- La actualización mas reciente del WAP Forum para el lenguaje de marcas utilizado en aplicaciones móviles es el XHTML-MP 2.0 el cual puede utilizarse para desarrollar tanto la versión web de un sitio como la móvil.

5.2. RECOMENDACIONES

- Para que la navegación del portal WAP sea optima, se recomienda que los dispositivos móviles en los que se desee utilizar la aplicación sean de tercera generación en adelante, debido a que estos tienen mejor capacidad para ejecutar cualquier aplicación móvil.

- Es recomendable cargar en el servidor lo estrictamente necesario para evitar problemas de sobrecarga ya que un portal WAP es limitado respecto al tamaño de imágenes y estilos.
- En la instalación del IIS 7.5 es recomendable utilizar el Pool de Aplicaciones Classic .Net, ya que, por defecto utiliza DefaultApp el cual está configurado para aplicaciones .NET.
- Para las pruebas de funcionalidad es recomendable utilizar MOBIREADY, esta herramienta de internet nos guía y nos permite seguir los estándares definidos por la W3C.
- La opinión de los usuarios que no están familiarizados con la navegación de un portal WAP es muy importante, esta es la razón principal por lo que las pruebas de usabilidad son recomendadas.

BIBLIOGRAFÍA

WEB:

- Maestros del Web. Redes Sociales en Internet.
<http://www.maestrosdelweb.com/editorial/redessociales/>. Consultado: 13 de junio de 2010.
- Vida digital. En la empresa 2.0, los trabajadores más conectados son los más rentables.
http://www.soitu.es/soitu/2008/02/19/pieldigital/1203419952_116938.html. Consultado: 13 de junio de 2010.
- Web Taller. Administración de proyectos.
<http://www.webtaller.com/maletin/articulos/administracion-proyectos-5.php>. Consultado: 13 de junio de 2010.
- Monografías. Sistemas operativos para redes y redes neuronales.
<http://www.monografias.com/trabajos15/sistemas-redes/sistemas-redes.shtml>. Consultado: 15 de junio de 2010.
- Wikipedia la enciclopedia libre. Estructura de descomposición del trabajo.
http://es.wikipedia.org/wiki/Estructura_de_descomposici%C3%B3n_del_trabajo. Consultado: 15 de junio de 2010.
- Scribd. FUNDAMENTOS DE LA METODOLOGIA RUP RATIONAL UNIFIED PROCESS. Juan Pablo Gomez. <http://www.scribd.com/doc/297224/RUP>. Consultado: 15 de junio de 2010.
- Scribd.. RUP. <http://www.scribd.com/doc/3968502/rup>. Consultado: 15 de junio de 2010.
- Especificación de Requerimientos.
<http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos>. Consultado: 22 de junio de 2010.
- Wikipedia la enciclopedia libre. Modelo de Negocio.
http://es.wikipedia.org/wiki/Modelo_de_negocio. Consultado: 22 de junio de 2010.

- Definición de Modelo de Negocio. <http://definicion.de/modelo-de-negocio/>. Consultado: 22 de junio de 2010.
- GestioPolis. Pasos para construir un diagrama de procesos. <http://www.gestiopolis.com/recursos/experto/catsexp/pagans/ger/36/procesos.htm> Consultado: 22 de junio de 2010.
- Diagramas de Operaciones de los Procesos. <http://148.202.148.5/cursos/id209/mzaragoza/unidad2/unidad2dos.htm>. Consultado: 22 de junio de 2010.
- Curso práctico de Modelado de Negocios con BPMN y UML. <http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>. Consultado: 29 de junio de 2010.
- Aprender Gratis. Introducción a BPM. <http://www.aprendergratis.com/introduccion-a-bpmn.html>. Consultado: 29 de junio de 2010.
- Visión general de la arquitectura de MySQL 5.1. <http://cnx.org/content/m18938/latest/>. Consultado: 29 de junio de 2010.
- Wikipedia la Enciclopedia Libre. Active Serve Pages. http://es.wikipedia.org/wiki/Active_Server_Pages. Consultado: 29 de junio de 2010.
- Maestros del Web. ¿Qué es ASP?. <http://www.maestrosdelweb.com/editorial/aspintro/>. Consultado: 10 de julio de 2010.
- Wikipedia la Enciclopedia Libre. JavaScript. <http://es.wikipedia.org/wiki/JavaScript>. Consultado: 10 de julio de 2010.
- Información sobre el Sistema WAP y sus funcionalidades. <http://www.melodiasmoviles.com/documentacion/wap.php>. Consultado: 15 de julio de 2010.
- Open Mobile Alliance. <http://www.wapforum.com/>. Consultado: 15 de julio de 2010.

- Red Gráfica Latinoamérica. Breve historia de los celulares y sus generaciones. <http://redgrafica.com/La-historia-de-los-celulares-y-sus>. Consultado: 17 de julio de 2010.
- Wikipedia la Enciclopedia Libre. Telefonía móvil 4G. http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G. Consultado: 17 de julio de 2010.
- Scribd. Manual de Bizagi. <http://www.scribd.com/doc/22659190/Manual-De-Bizagi>. Consultado: 17 de julio de 2010.
- Monografías. Manual de procedimiento. <http://www.monografias.com/trabajos13/mapro/mapro.shtml>. Consultado: 18 de julio de 2010.
- Tecnologías Móviles para los próximos años. <http://www.xatakamovil.com/futuro/tecnologias-moviles-para-los-proximos-anos>. Consultado: 18 de julio de 2010.
- Especificación de Requerimientos de Software (SRS). <http://readysset.tigris.org/nonav/es/templates/srs.html>. Consultado: 18 de julio de 2010.
- Wikipedia la Enciclopedia Libre. Proceso Unificado de Rational. http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational. Consultado: 30 de julio de 2010.
- Wikipedia la Enciclopedia Libre. Lenguaje Unificado de Modelado
- http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado. Consultado: 30 de julio de 2010.
- Getting Started with UML. <http://www.uml.org/>. Consultado: 30 de julio de 2010.
- Tutorial de UML. <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>. Consultado: 31 de julio de 2010.
- Modelado de Sistemas con UML. <http://lucas.hispalinux.es/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>. Consultado: 31 de julio de 2010.

- Monografias. Ejemplos de diagramas UML, interfaces gráficas de usuario, y usos del UML en la ingeniería inversa.
<http://www.monografias.com/trabajos67/diagramas-uml/diagramas-uml.shtml>.
Consultado: 02 de agosto de 2010.

LIBROS:

- UML y Patrones, de Craig Larman. Consultado:30 de julio de 2010.
- El Lenguaje Unificado de Modelado, de Grady Booch, James Rumbaugh, Ivar Jacobson. Consultado: 1 de Agosto del 2010.
- Fundamentos De Programación En Asp 3.0, de Viguer, J. M. Consultado:10 de julio del 2010.
- PMPBook, de Kim Heldman, Consultado:30 de junio del 2010.